

# ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

Τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών



## Μεταπτυχιακή Διατριβή

Διαχείριση Γεωγραφικών Ερωτημάτων και  
Υπηρεσία Διαδραστικών Χαρτών για  
Ιστοσελίδες

Geospatial Query Management and  
Interactive-Maps Service for Web Pages

Πολυχρόνης Βελέντζας  
ΑΕΜ 00517

Επιβλέπων Καθηγητής  
Μιχαήλ Βασιλακόπουλος

Βόλος, Ιούνιος 2017



© ΠΘ, 2017

Η παρούσα μεταπτυχιακή διατριβή, η οποία εκπονήθηκε στο πλαίσιο του Προγράμματος Μεταπτυχιακών Σπουδών (Π.Μ.Σ.) «Επιστήμη και Τεχνολογία Ηλεκτρολόγου Μηχανικού και Μηχανικού Υπολογιστών», και τα λοιπά αποτελέσματα της μεταπτυχιακής διατριβής αποτελούν συνιδιοκτησία του Πανεπιστημίου Θεσσαλίας και του μεταπτυχιακού φοιτητή, ο καθένας από τους οποίους έχει το δικαίωμα ανεξάρτητης χρήσης και αναπαραγωγής τους (στο σύνολο ή τμηματικά) για διδακτικούς και ερευνητικούς σκοπούς, σε κάθε περίπτωση αναφέροντας τον τίτλο και τον συγγραφέα και το Πανεπιστήμιο Θεσσαλίας, όπου εκπονήθηκε η μεταπτυχιακή διατριβή, καθώς και τον επιβλέποντα και την Επιτροπή Κρίσης.



**Βελέντζας Πολυχρόνης**  
**Διαχείριση Γεωγραφικών Ερωτημάτων και**  
**Υπηρεσία Διαδραστικών Χαρτών για Ιστοσελίδες**



## ΕΥΧΑΡΙΣΤΙΕΣ

---

Στα πλαίσια της εκπόνησης της μεταπτυχιακής μου διατριβής κλήθηκα να υλοποιήσω μια εφαρμογή που αφορά τον σχεδιασμό και προβολή γεωγραφικών ερωτημάτων. Προχωρήσαμε στο σχεδιασμό και την υλοποίησή της με τον επιβλέποντα καθηγητή κ. Βασιλακόπουλο Μιχαήλ του οποίου η βοήθειά στην ολοκλήρωση της μεταπτυχιακής μου διατριβής ήταν ουσιαστική και αποδείχθηκε πολύτιμος συνεργάτης τόσο με τις ιδέες του όσο και με την καθοδήγησή του. Το ήθος του και η προσήλωση του στα εκπαιδευτικά του καθήκοντα είναι άξια συγχαρητηρίων και ως φυσικό επακόλουθο η διαδικασία της μάθησης στο πλευρό του είναι ένα μαγευτικό ταξίδι.

Ευχαριστώ την οικογένειά μου που με υποστήριξε και κυρίως την σύζυγό μου Ευτυχία που σήκωσε το μεγαλύτερο οικογενειακό βάρος, τις ώρες της απουσίας μου. Ένα μεγάλο ευχαριστώ στα παιδιά μου Ανδρομάχη και Κωστή που έδειξαν κατανόηση και ελπίζω να τους αποζημιώσω τις μέρες που έπονται, αφιερώνοντάς τους τον χρόνο που τους αξίζει.



**Βελέντζας Πολυχρόνης**  
**Διαχείριση Γεωγραφικών Ερωτημάτων και**  
**Υπηρεσία Διαδραστικών Χαρτών για Ιστοσελίδες**



## “Διαχείριση Γεωγραφικών Ερωτημάτων και Υπηρεσία Διαδραστικών Χαρτών για Ιστοσελίδες”

Βελέντζας Πολυχρόνης

Βασιλακόπουλος Μιχαήλ  
Τμήμα Ηλεκτρολόγων  
Μηχανικών και Μηχανικών  
Υπολογιστών  
Πανεπιστήμιο Θεσσαλίας

Βάβαλης Εμμανουήλ  
Τμήμα Ηλεκτρολόγων  
Μηχανικών και Μηχανικών  
Υπολογιστών  
Πανεπιστήμιο Θεσσαλίας

Τσομπανοπούλου Παναγιώτα  
Τμήμα Ηλεκτρολόγων  
Μηχανικών και Μηχανικών  
Υπολογιστών  
Πανεπιστήμιο Θεσσαλίας

Οι σύγχρονες ανάγκες πληροφόρησης με πλούσιο περιεχόμενο, ενσωματωμένο σε ψηφιακούς χάρτες ολοένα και αυξάνονται. Οι γεωγραφικές και γεωχρονικές (πολλές φορές αναφέρονται και ως χωροχρονικές) πληροφορίες που συγκεντρώνονται από τις αμέτρητες πλέον ψηφιακές συσκευές έχουν δημιουργήσει την ανάγκη πληρέστερης και αποδοτικότερης αξιοποίησης και προβολής τους.

Αυτή η μεταπτυχιακή διατριβή παρουσιάζει μια πλατφόρμα η οποία αυτοματοποιεί την πολύπλοκη και σύνθετη διαδικασία της συλλογής των γεωγραφικών και γεωχρονικών δεδομένων, την αποθήκευσή τους και τελικά την προβολή τους μέσω του διαδικτύου σε οποιαδήποτε συσκευή. Έχει δοθεί ιδιαίτερη έμφαση στον ανοιχτό αρχιτεκτονικό σχεδιασμό της πλατφόρμας, ούτως ώστε να μπορεί να την χρησιμοποιήσει κανείς είτε αυτούσια, είτε συνδυάζοντάς την με τις δικές του εφαρμογές πολύ εύκολα και αποδοτικά.

Αποτελείται από ένα σύνολο εφαρμογών που η καθεμία έχει σχεδιασθεί να λειτουργεί αρμονικά με τις υπόλοιπες και να εξυπηρετεί ένα συγκεκριμένο σκοπό. Το βασικό συστατικό της πλατφόρμας είναι το εργαλείο «Σχεδίασης Γεωγραφικών Ερωτημάτων», όπου ο διαχειριστής παραμετροποιεί και αποθηκεύει τα ερωτήματά του ενώ παράλληλα οπτικοποιεί το αποτέλεσμα τους. Για τις ανάγκες παρουσίασης των ερωτημάτων σε φορητές και μη συσκευές έχει αναπτυχθεί η «Υπηρεσία Διαδραστικών Χαρτών», όπου χωρίς κώδικα μπορούμε να εμφανίσουμε τα αποθηκευμένα ερωτήματά μας. Απαραίτητο συμπλήρωμα της πλατφόρμας αποτελεί η «Υπηρεσία Γεωγραφικών Δεδομένων REST API», όπου μας επιτρέπει την σύνδεση εξωτερικών εφαρμογών, τόσο για άντληση δεδομένων όσο και για την αποθήκευσή τους.

Για να δοκιμάσουμε την λειτουργία της πλατφόρμας, επιπλέον αναπτύχθηκε μια εφαρμογή για κινητές συσκευές Android, iPhone και iPad, η οποία συλλέγει γεωχρονικά δεδομένα και τα αποθηκεύει απομακρυσμένα με την βοήθεια του REST API. Αυτά τα γεωχρονικά δεδομένα στη συνέχεια παρουσιάζονται στον τελικό χρήστη με τις αντίστοιχες προβολές της «Υπηρεσίας Διαδραστικών Χαρτών».

Λέξεις-Κλειδιά : Γεωγραφικά Συστήματα Πληροφοριών, Χάρτης, Γεωγραφικά Ερωτήματα, Γεωχρονικά Δεδομένα, Κινητές Συσκευές, Smartphones, REST API, Embarcadero Delphi



*Βελέντζας Πολυχρόνης*  
*Διαχείριση Γεωγραφικών Ερωτημάτων και*  
*Υπηρεσία Διαδραστικών Χαρτών για Ιστοσελίδες*



## “Geospatial Query Management and Interactive-Maps Service for Web Pages”

Velentzas Polychronis

**Vasilakopoulos Michael**  
Department of Electrical and  
Computer Engineering  
University of Thessaly  
Volos, Greece

**Vavalis Emmanouil**  
Department of Electrical  
and Computer Engineering  
University of Thessaly  
Volos, Greece

**Tsompanopoulou Panagiota**  
Department of Electrical and  
Computer Engineering  
University of Thessaly  
Volos, Greece

Modern, content rich information requirements, embedded in digital maps, are increasing rapidly. Geographic and geo-temporal data are being produced from countless digital devices, rising the need for more complete and efficient means of evaluation and projection usage.

This paper presents a platform that automates the complex and perplexed procedure of collecting geographic and geo-temporal data, storing the data and finally presenting them to any internet enabled device that comes with a modern browser. Special emphasis has been placed on the open architectural design of the platform, so that it can be used either by itself or by combining it with custom applications very easily and efficiently.

It consists of a set of applications that each is designed to work in seamlessly with the rest and each one serves a particular purpose. The basic component of the platform is the "Geographic Query Design" tool, where the administrator configures and stores his queries and in the same time he visualizes their results. For the needs of presenting queries on portable or not devices, the "Interactive Maps Service" has been developed, where we can display our stored queries without a single line of code. Furthermore, a necessary complement to the platform is the "REST API Geographical Data Service", which allows us to connect external applications, both for data retrieval and for storage.

To test the platform's functionality, an Android, iPhone, and iPad mobile application has been developed to collect geo-temporal data and store them remotely using the REST API. These geo-temporal data are then presented to the end user with the corresponding views of the "Interactive Maps Service".

**Keywords:** Geographic Information Systems, Map, Geographic Query, Geo-temporal data, Mobile Devices, Smartphones, REST API, Embarcadero Delphi





*Βελέντζας Πολυχρόνης*  
*Διαχείριση Γεωγραφικών Ερωτημάτων και*  
*Υπηρεσία Διαδραστικών Χαρτών για Ιστοσελίδες*



## Περιεχόμενα

---

1.	Εισαγωγή .....	11
1.1.	Σκοπός .....	11
1.2.	Ανασκόπηση Γεωγραφικών Συστημάτων Πληροφοριών .....	12
1.3.	Δομή Κεφαλαίων .....	24
2.	Δομή Πλατφόρμας .....	27
2.1.	Βάση Δεδομένων .....	28
2.2.	Ιστοσελίδα .....	31
2.3.	Σχεδιασμός Γεωγραφικών Ερωτημάτων .....	32
2.4.	Υπηρεσία Διαδραστικών Χαρτών .....	32
2.5.	Υπηρεσία Γεωγραφικών Δεδομένων REST API .....	32
2.6.	Εφαρμογή Κινητού, Συλλογή Δεδομένων GPS .....	33
3.	Εργαλεία Ανάπτυξης και Προγραμματισμού .....	35
3.1.	PostgreSQL 9.5, PostGIS 2.3 Extension .....	35
3.2.	Embarcadero Delphi XE 10.2 .....	37
3.3.	FMSoft Unigui .....	39
3.4.	Sencha Ext Js, Sencha Touch .....	41
3.5.	NGINX 1.11.11 .....	42
3.6.	Google Map API v3 .....	45
3.7.	Insomnia Rest Client v5 .....	49
4.	Παρουσίαση Χρήσης Πλατφόρμας .....	85
4.1.	Ιστοσελίδα .....	85
4.2.	Διαχείριση Ερωτημάτων .....	88
4.3.	Υπηρεσία Διαδραστικών Χαρτών .....	99
4.4.	Υπηρεσία Γεωγραφικών Δεδομένων REST API .....	101



4.5. Εφαρμογή κινητού, Συλλογή δεδομένων GPS .....	108
5. Ανάπτυξη Πλατφόρμας.....	51
5.1. Maps .....	52
5.2. MapsViewer .....	68
5.3. RestService.....	72
5.4. TrackLocation .....	77
5.5. Nginx Reverse Proxy .....	81
6. Αξιολόγηση και Συμπεράσματα .....	85



## 1. Εισαγωγή

Στα πλαίσια της εκπόνησης της μεταπτυχιακής μου διατριβής κλήθηκα να υλοποιήσω μια εφαρμογή που αφορά τον σχεδιασμό και προβολή γεωγραφικών ερωτημάτων. Σαν στόχο είχαμε την δημιουργία μιας χρηστικής εφαρμογής η οποία θα μπορούσε να σταθεί επάξια στον χώρο των γεωγραφικών συστημάτων.

### 1.1. Σκοπός

Ο σκοπός της μεταπτυχιακής διατριβής είναι να υλοποιηθεί μια πλατφόρμα στην οποία θα μπορεί ένας διαχειριστής να σχεδιάσει μια γεωγραφική βάση δεδομένων και να την χρησιμοποιήσει για να προβάλει γεωγραφικά ερωτήματα μέσω του διαδικτύου σε οποιαδήποτε συσκευή με σύγχρονο φυλλομετρητή. Υπάρχουν αρκετές εφαρμογές εμπορικές και μη οι οποίες προσεγγίζουν το συγκεκριμένο τομέα, αλλά υστερούν στην προβολή των γεωγραφικών ερωτημάτων, μέσω του διαδικτύου, στοχεύοντας σε ένα ευρύτερο κοινό. Επιπλέον, δεν υπάρχει στις υπάρχουσες υλοποιήσεις, ένας εύκολος με ανοιχτή αρχιτεκτονική τρόπος σύνδεσης εξωτερικών εφαρμογών, για τις ανάγκες αποθήκευσης ή ανάκτησης γεωγραφικών δεδομένων.

Ο σχεδιασμός σε αυτή την πλατφόρμα, όπως θα δούμε παρακάτω, μπορεί να απαιτεί ορισμένες βασικές γνώσεις γεωγραφικών ερωτημάτων από την πλευρά του διαχειριστή, αλλά μπορεί να δημιουργήσει χάρτες με το περιεχόμενο των γεωγραφικών ερωτημάτων χωρίς επιπλέον κώδικα. Εκτός από την αυτόματη δημιουργία των τελικών διαδραστικών χαρτών, ένας προγραμματιστής θα μπορούσε να χρησιμοποιήσει τις υπηρεσίες της πλατφόρμας για να τις ενσωματώσει στις εφαρμογές του ή στην ιστοσελίδα του.

Για να αναδείξουμε το υπάρχον πρόβλημα της διαχείρισης γεωγραφικών ή γεωχρονικών δεδομένων και την παρουσίασής τους σε μια ιστοσελίδα θα πρέπει να σκεφτούμε το παρακάτω παράδειγμα. Έστω ότι υπάρχει η ανάγκη παρακολούθησης ενός στόλου οχημάτων τόσο σε πραγματικό χρόνο όσο και σε παρελθόντα χρόνο. Επιπλέον το προς σχεδίαση σύστημα θα απαιτεί συχνές αλλαγές στις προβολές και στα γεωγραφικά ερωτήματα. Σε αυτή την περίπτωση ο προγραμματιστής θα πρέπει να κάνει τα εξής:

1. Εγκατάσταση web server σε διαδικτυακό server.
2. Εγκατάσταση βάσης δεδομένων, σε αυτόν τον server με δυνατότητα γεωγραφικών δεδομένων
3. Δημιουργία διαχειριστικής ιστοσελίδας με απαραίτητη διαδικασία authentication
4. Δημιουργία υπηρεσία τύπου REST, για την συλλογή και ανάκτηση δεδομένων
5. Ανάπτυξη εφαρμογής για συλλογή στίγματος GPS και αποστολή στην υπηρεσία REST



6. Δημιουργία σελίδων για προβολή γεωγραφικών ερωτημάτων, με τεχνολογίες όπως Google Maps
7. Διαχειριστικές διαδικασίες της βάσης δεδομένων (backup, restore)

Είναι προφανές ότι ο χρόνος ανάπτυξης ενός τέτοιου συστήματος θα είναι εξαιρετικά μεγάλος και λόγω της πολυπλοκότητας του συστήματος, το αποτέλεσμα πιθανώς δεν θα είναι το καλύτερο δυνατό. Επιπλέον οποιαδήποτε αλλαγή στο σύστημα, όπως προσθήκη μιας νέας προβολής ενός νέου γεωγραφικού ή γεωχρονικού ερωτήματος θα ήταν μια επίπονη και χρονοβόρα διαδικασία.

Με την πλατφόρμα που παρουσιάζουμε θα δούμε ότι η παραπάνω πολυπλοκότητα περιορίζεται στον σχεδιασμό της βάσης δεδομένων και την δημιουργία των κατάλληλων γεωγραφικών, γεωχρονικών ερωτημάτων.

## 1.2. Ανασκόπηση Γεωγραφικών Συστημάτων Πληροφοριών

Προτού παρουσιάσουμε την πλατφόρμα μας, κρίνεται αναγκαία η ανάλυση λειτουργίας των υπαρχόντων γεωγραφικών συστημάτων πληροφοριών, προκειμένου να μελετήσουμε την λειτουργία τους, την χρηστικότητα και την αποτελεσματικότητά τους. Υπάρχουν αρκετές εφαρμογές γεωγραφικών συστημάτων ανοικτού κώδικα και μη. Στην συνέχεια θα περιγράψουμε τα παρακάτω συστήματα και τις λειτουργίες τους.

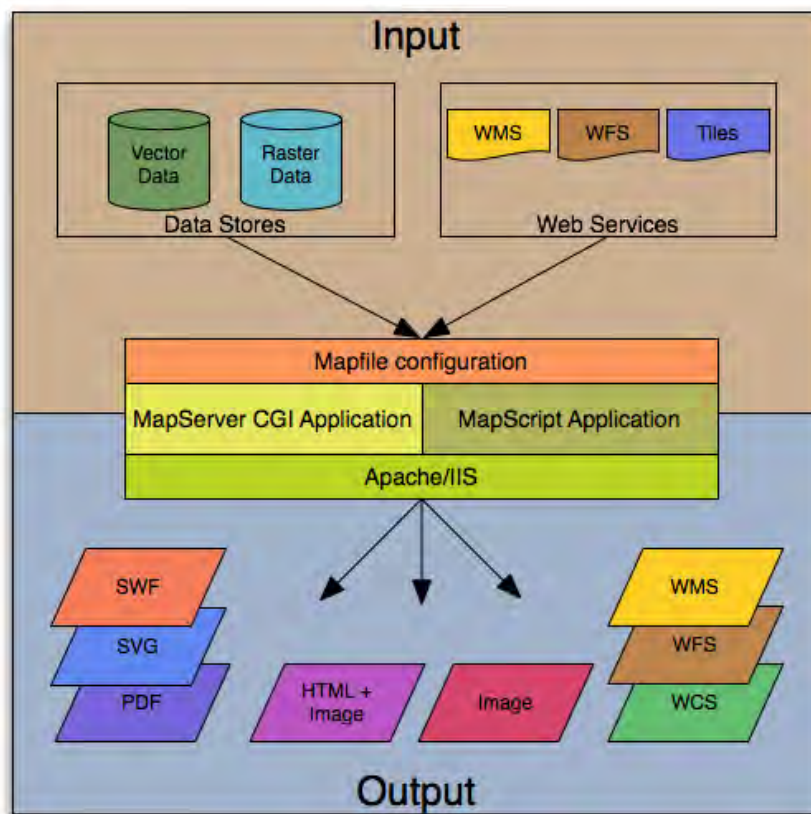
1. MapServer
2. uDig
3. Grass GIS
4. Esri ArcGIS
5. QGIS
6. OpenStreetMap

### MapServer [1]

Η εφαρμογή MapServer είναι μια από τις πιο δημοφιλείς εφαρμογές προβολής δυναμικών γεωγραφικών δεδομένων μέσω διαδικτύου. Έχει πλούσια χαρακτηριστικά όπως :

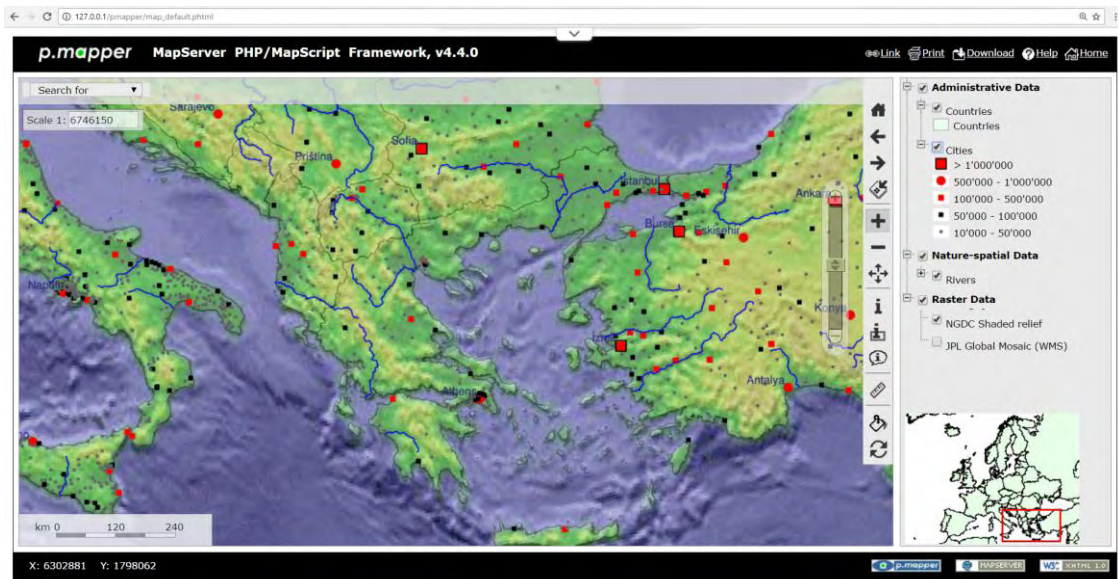
- ❖ Υποστηρίζει γεωγραφικά ερωτήματα σε δεδομένα Raster, Vector και διαφόρων Database formats και τα προβάλει.
- ❖ Εκτελείται σε όλα τα λειτουργικά συστήματα.
- ❖ Υποστηρίζει script γλώσσες όπως PHP, Python, Perl, Ruby, Java, .Net.
- ❖ Προβολές on-the-fly
- ❖ Υψηλής ποιότητας rendering.

Στην πιο βασική του μορφή, το MapServer είναι ένα CGI πρόγραμμα το οποίο περιμένει ανενεργό στον Web Server. Όταν έρθει μια κλήση, αυτή προωθείται στο MapServer και αυτό χρησιμοποιεί τις παραμέτρους της κλήσης και ένα αρχείο χάρτη (Mapfile) για να δημιουργήσει μια εικόνα. Επιπλέον θα μπορούσε να επιστρέψει και εικόνες, σχετικούς χάρτες, κλίμακες και λοιπές τιμές οι οποίες τροφοδοτήθηκαν ως μεταβλητές CGI.



Εικόνα 1.2.1 Ανατομία μιας εφαρμογής MapServer

Η εγκατάστασή του είναι αρκετά εύκολη και το documentation πλήρες. Δοκιμάζοντας το MapServer καταφέραμε να παρουσιάσουμε δυναμικούς γεωγραφικούς χάρτες όπως φαίνεται στην παρακάτω εικόνα :



Εικόνα 1.2.2 Χάρτης MapServer.

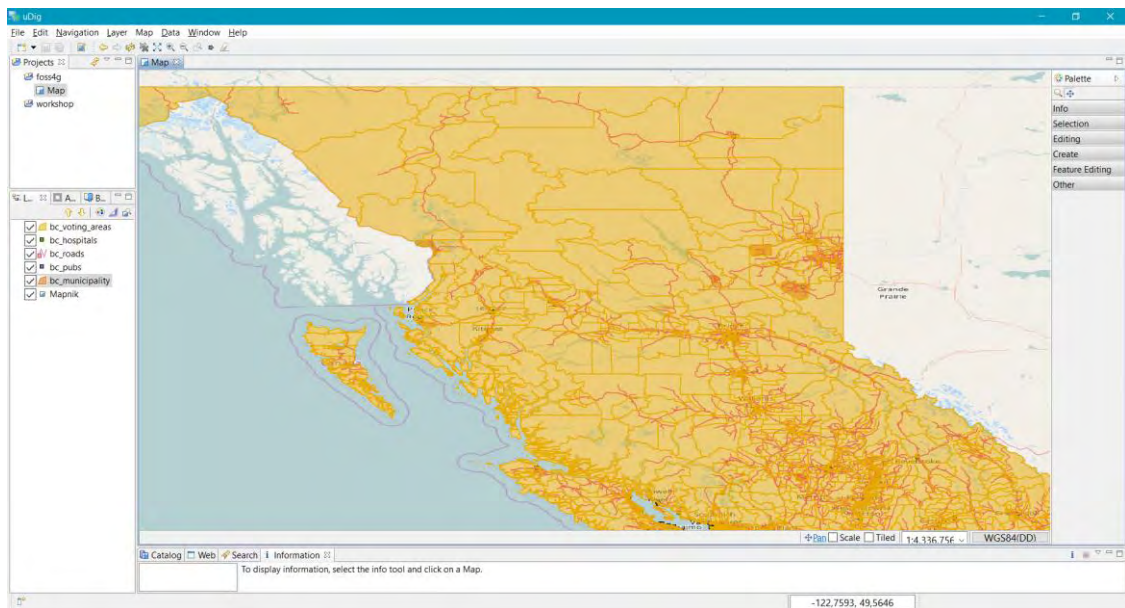
## uDig [2]

Το uDig είναι μια ανοικτού κώδικα εφαρμογή για επιτραπέζιους υπολογιστές και διανέμεται με άδειες EPL και BSD. Το uDig παρέχει:

- ❖ Μια out-of-the-box εφαρμογή με την δυνατότητα να προβάλει, επεξεργαστεί και να τυπώσει, από βάσεις δεδομένων όπως Oracle, SDE, PostGIS καθώς επίσης και από Web Feature Servers (WFS).
- ❖ Εφαρμόζει το στάνταρντ extension framework RPC, με το οποίο μπορούμε να προσθέσουμε δυνατότητες και λειτουργίες στην εφαρμογή.
- ❖ Μια πλήρη εργαλειοθήκη με λειτουργίες Java, έτοιμες για χρήση από τους προγραμματιστές
- ❖ Native υποστήριξη για Windows, Mac OS/X και Linux.

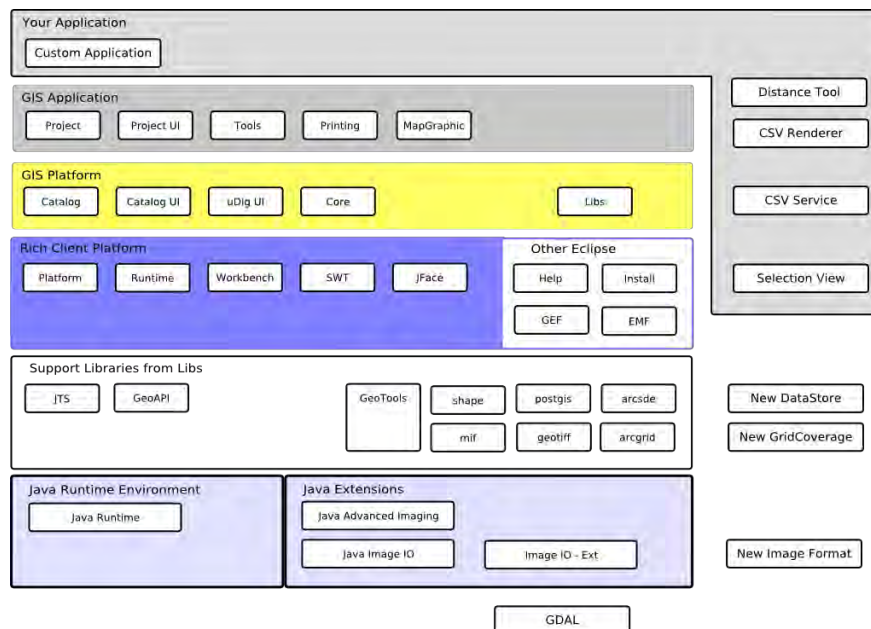
Το αποτέλεσμα της εφαρμογής είναι αρκετά ποιοτικό με πολλές δυνατότητες παραμετροποίησης.





Εικόνα 1.2.3 Χάρτης uDig

Δοκιμάζοντας το uDig μας έδωσε την αίσθηση ενός δυνατού εργαλείου χωρίς όμως την δυνατότητα να παρουσιάσουμε δυναμικούς χάρτες στο διαδίκτυο. Μπορεί όμως να χρησιμοποιηθεί ως plug-in στο περιβάλλον προγραμματισμού Eclipse. Η αρχιτεκτονική που χρησιμοποιεί εμφανίζεται στην παρακάτω εικόνα:

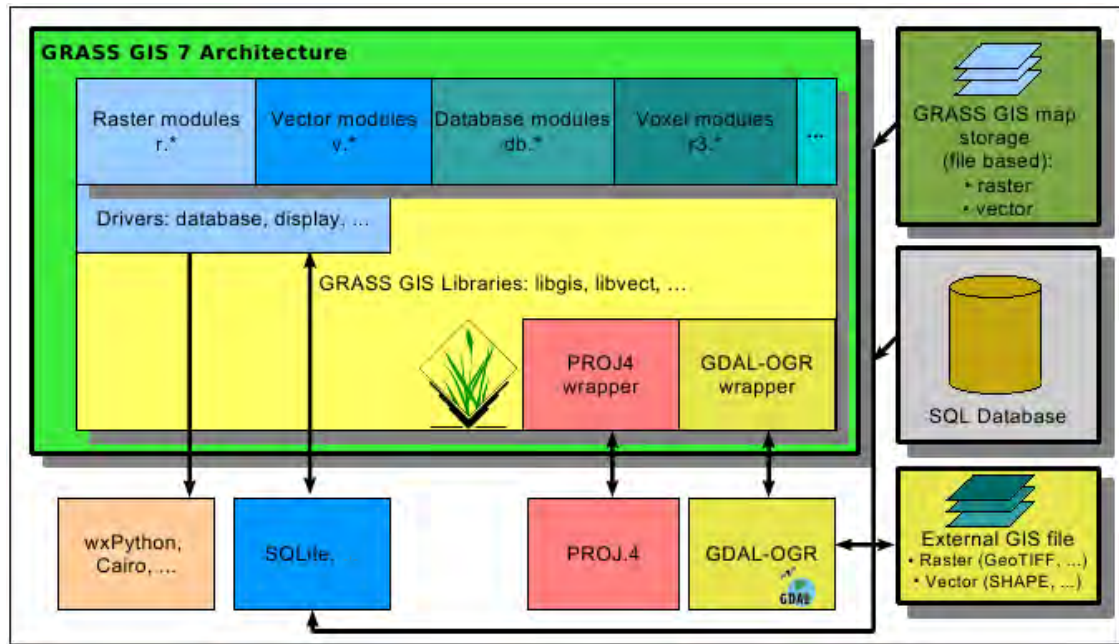


Εικόνα 1.2.4 Αρχιτεκτονική uDig

Grass GIS [3]



Το GRASS (Geographic Resources Analysis Support System), είναι ένα γεωγραφικό σύστημα πληροφοριών (GIS) που χρησιμοποιείται για διαχείριση δεδομένων, επεξεργασία εικόνας, παραγωγή γραφικών, γεωγραφική μοντελοποίηση και οπτικοποίηση πληθώρας τύπων δεδομένων. Είναι ελεύθερο λογισμικό με άδεια GPL. Η αρχιτεκτονική του συνοψίζεται στο παρακάτω σχέδιο :

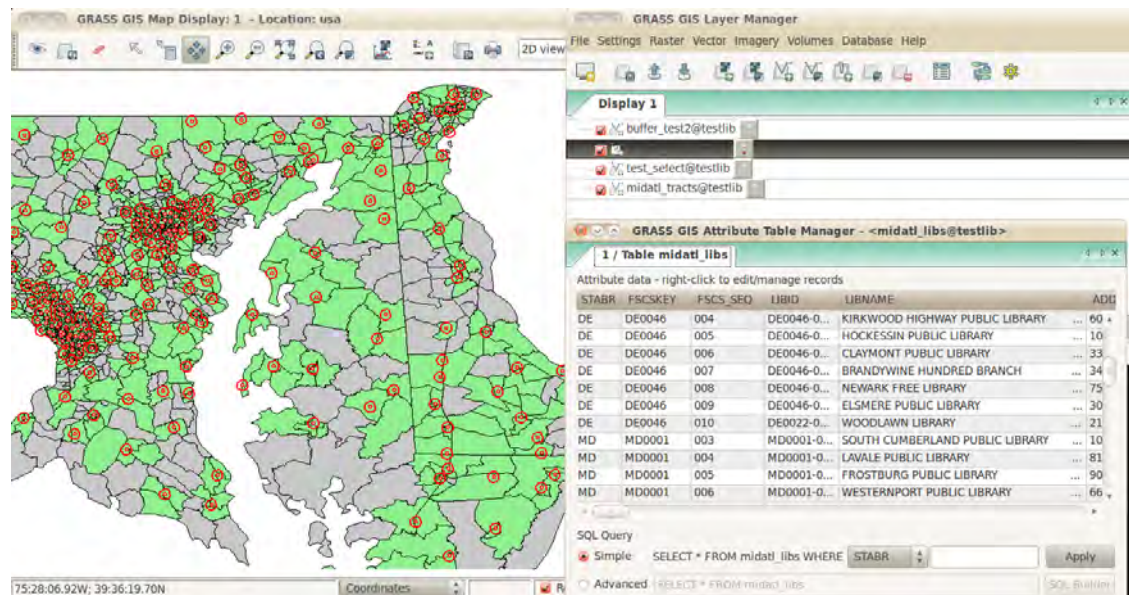


Εικόνα 1.2.5 Αρχιτεκτονική Grass GIS

Αναλυτικότερα, περιλαμβάνει τα παρακάτω χαρακτηριστικά:

- ❖ Raster Analysis
- ❖ 3D-Raster (voxel) Analysis
- ❖ Vector Analysis
- ❖ Point data analysis
- ❖ Image processing
- ❖ DTM-Analysis
- ❖ Geocoding
- ❖ Visualization: 3D επιφάνειες με 3D query (NVIZ)
- ❖ Map creation: Image maps, Postscript maps, HTML maps
- ❖ SQL-support: Database interfaces (DBF, SQLite, PostgreSQL, MySQL, ODBC)
- ❖ Geostatistics: Interface για "R" (περιβάλλον στατιστικής ανάλυσης), Matlab κτλ
- ❖ Temporal framework: χρονοχωρική ανάλυση
- ❖ Erosion modelling, Landscape structure analysis, Solution transport, Watershed analysis.

Δείγμα παραγόμενου χάρτη εμφανίζεται παρακάτω :



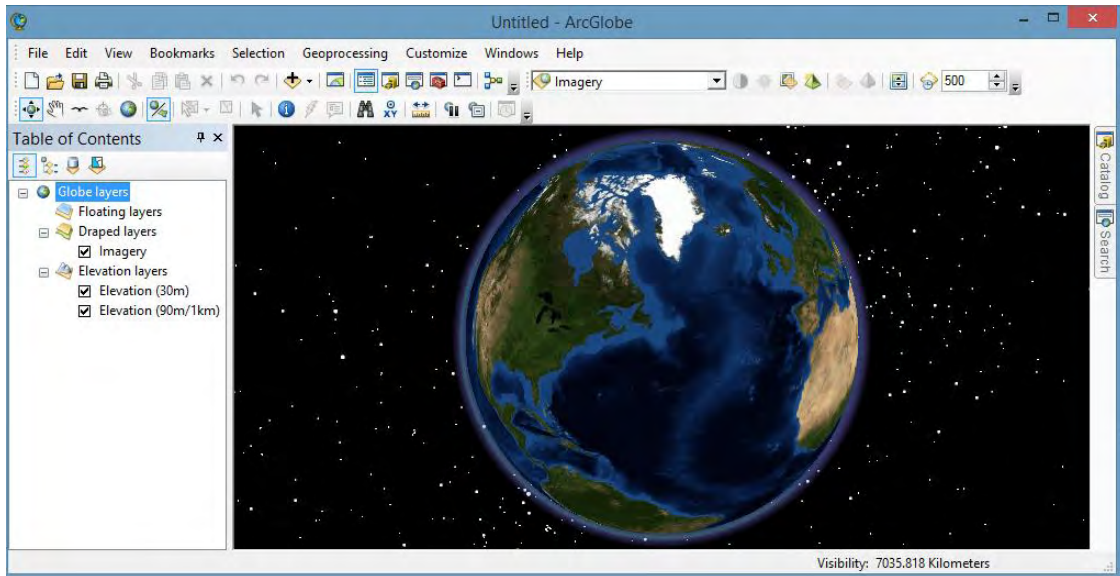
Εικόνα 1.2.6 Χάρτης Grass GIS

#### Esri ArcGIS [4]

Το Esri ArcGIS είναι ένα από τα πιο ολοκληρωμένα GIS συστήματα. Τα πιο δυνατά του χαρακτηριστικά αφορούν την επεξεργασία, αποθήκευση, ανάλυση και οπτικοποίηση γεωγραφικών δεδομένων. Τα καινοτόμα του χαρακτηριστικά είναι πολλά και η εφαρμογή είναι εξαιρετικά συμπαγής, μιας και η έναρξή της ξεκινά από την δεκαετία του '70. Το συγκεκριμένο GIS σύστημα έχει αναβαθμιστεί αρκετές φορές, και έχει εμπλουτισθεί με πολλά νέα χαρακτηριστικά. Οι δυνατότητές του ενισχύονται με τις παρακάτω επεκτάσεις :

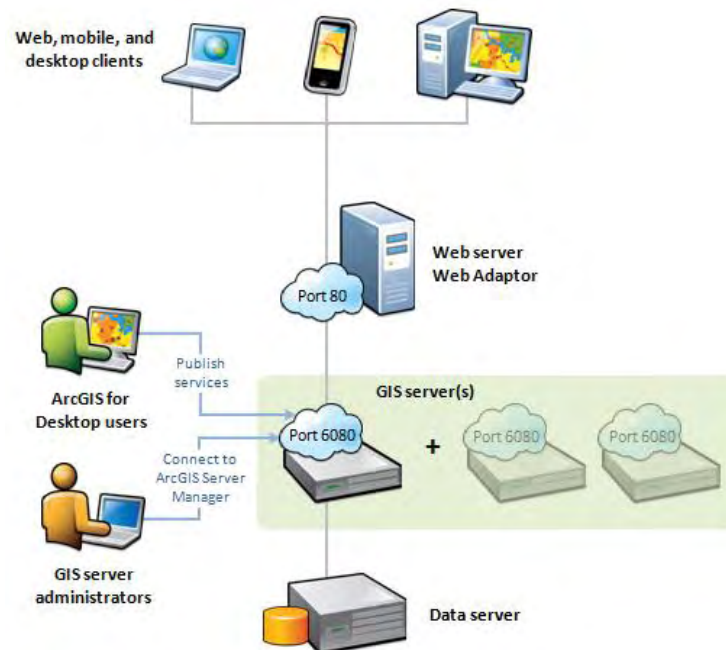
- ❖ 3D Analyst extension: περιέχει εργαλεία για μοντελοποίηση επιφανειών και τρισδιάστατη απεικόνιση.
- ❖ ArcScan extension: υποστηρίζει την αυτόματη δημιουργία vector features από raster εικόνες.
- ❖ Geostatistical Analysis extension: παρέχει μεγάλη γκάμα από εργαλεία γεωγραφικής ανάλυσης.
- ❖ Network Analysis tools: Υποστήριξη network datasets
- ❖ Publisher extension: Δημιουργία PMF αρχείων.
- ❖ Schematics extensions: Δημιουργία σχηματικών διαγραμμάτων από network data αποθηκευμένα σε geodatabases.
- ❖ Spatial Analyst extension: Γεωγραφική ανάλυση με raster δεδομένα.

- ❖ The Tracking Analyst extension: Χωροχρονική ανάλυση με ιστορικά και πραγματικού χρόνου δεδομένα.



Εικόνα 1.2.7 Χάρτης ArcGIS

### ArcGIS Server site architecture

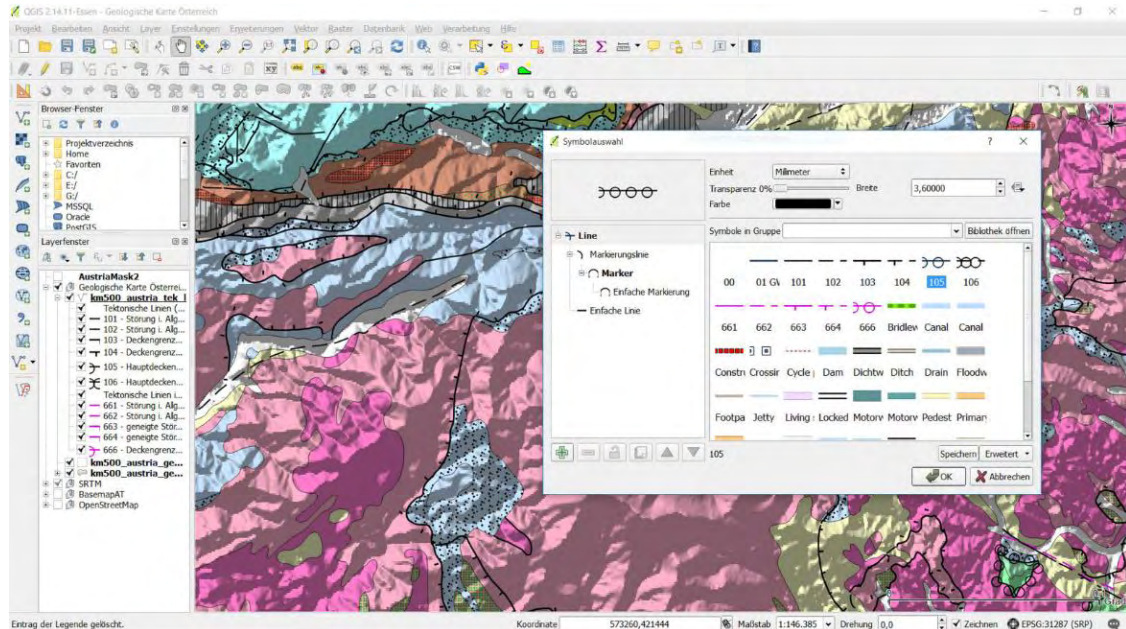


Εικόνα 1.2.8 Αρχιτεκτονική ArcGIS

QGIS [5]



Το QGIS θεωρείται το καλύτερο GIS εργαλείο στην κοινότητα του δωρεάν ανοικτού κώδικα (FOSS: free and open source software). Προσφέρει πολλές από τις κοινές λειτουργίες εγγενώς ως λειτουργίες του πυρήνα του αλλά και με διάφορα πρόσθετα plugin.



Εικόνα 1.2.9 Περιβάλλον QGIS

Παρακάτω θα περιγράψουμε τα πιο σημαντικά χαρακτηριστικά του QGIS, καθώς και μια περιγραφή της κονσόλας python που διαθέτει.

#### ❖ View Data

Η εισαγωγή επιπέδων vector και raster δεδομένων διαφορετικών format, πραγματοποιείται χωρίς την ανάγκη να μετατραπούν σε ένα εσωτερικό format. Τα υποστηριζόμενα format είναι τα:

- Spatial Databases, PostGIS, Spatialite και MS SQL Spatial, Oracle Spatial
- Format raster και imagery, όπως GeoTIFF, ERDAS IMG, ArcInfo ASCII GRID, JPEG, PNG και άλλα
- GRASS raster και vector δεδομένα από GRASS βάσεις δεδομένων
- Online γεωγραφικά δεδομένα από OGC Web Services, περιλαμβανομένων τα WMS, WMTS, WCS, WFS, and WFS-T

#### ❖ Explore data and compose maps

Η υποστήριξη σύνθεσης χαρτών, επιτρέπει με διαδραστικό τρόπο την εξερεύνηση των γεωγραφικών δεδομένων με φιλικό προς τον χρήστη τρόπο. Εργαλεία που περιλαμβάνει το GUI του είναι τα:

- QGIS browser
- On-the-fly reprojection
- DB Manager



- Map composer
- Overview panel
- Spatial bookmarks
- Annotation tools
- Identify/select features
- Edit/view/search attributes
- Data-defined feature labelling
- Data-defined vector and raster symbology tools
- Atlas map composition with graticule layers
- North arrow scale bar and copyright label for maps
- Support for saving and restoring projects
- ❖ Create, edit, manage and export data  
Υποστήριξη δημιουργίας, επεξεργασίας και εξαγωγής vector και raster επίπεδα (layers) σε πολλά format. Επιπλέον προσφέρει τα:
  - Digitizing εργαλεία για OGR-supported formats και GRASS vector layers
  - Ικανότητα να δημιουργεί και επεξεργασθεί shapefiles και GRASS vector layers
  - Georeferencer plugin για geocode εικόνες
  - GPS εργαλεία για εισαγωγή και εξαγωγή GPX format, και μετατροπή άλλων GPS formats σε GPX
  - Υποστήριξη OpenStreetMap data
  - Δημιουργία spatial database tables από shapefiles με το DB Manager plugin
  - Εξελιγμένη διαχείριση των spatial database tables
  - Εργαλεία για διαχείριση vector attribute tables
  - Επιλογή για αποθήκευση screenshots σαν georeferenced εικόνες
  - DXF-Export εργαλείο με εμπλουτισμένες δυνατότητες για εξαγωγή στυλ και plugins για εκτέλεση CAD-like εργασίες
- ❖ Analyze Data  
Δυνατότητα γεωγραφικών αναλύσεων δεδομένων σε γεωγραφικές βάσεις δεδομένων και άλλα OGR format. Το QGIS προσφέρει εργαλεία vector analysis, sampling, geoprocessing, geometry και database management.
- ❖ Publish maps on the Internet  
Δυνατότητα χρησιμοποίησης σαν WMS, WMTS, WMS-C ή WFS και WFS-T client, και σαν WMS, WCS ή WFS server.
- ❖ Extend QGIS functionality through plugins  
Προσαρμογή αναγκών του χρήστη με την χρήση αρχιτεκτονικής επεκτάσεων με plugin καθώς και με βιβλιοθήκες που μπορούν να δημιουργήσουν plugin. Είναι δυνατή η δημιουργία νέων εφαρμογών με C++ ή Python.
- ❖ Python console

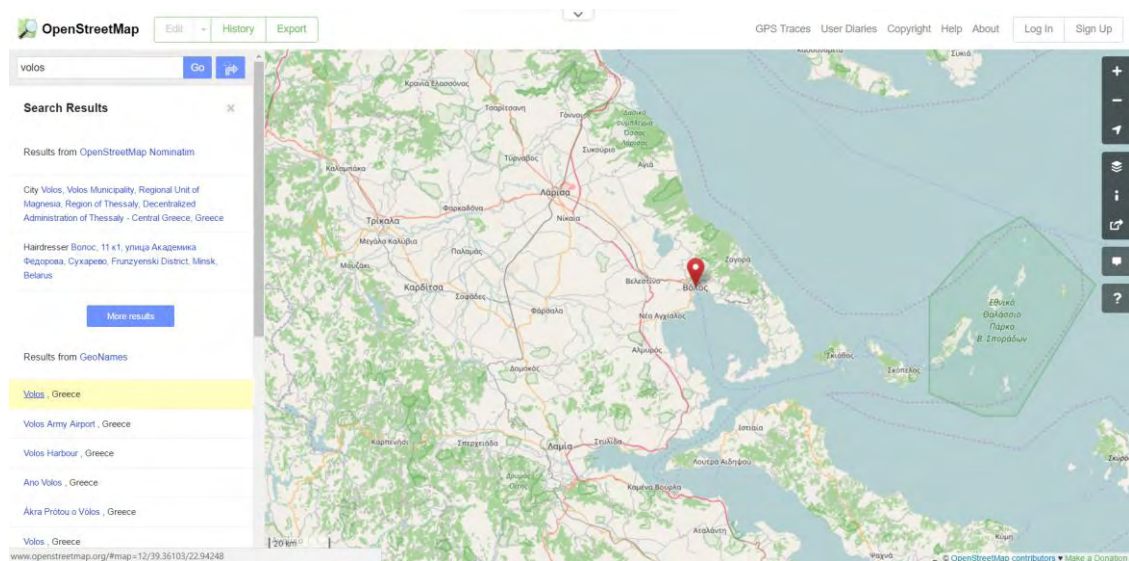
Υποστήριξη κονσόλας Python, όπου μπορεί κανείς να γράψει υψηλού επιπέδου κώδικα script της Python.

### OpenStreetMap [6]

Το OpenStreetMap έχει δημιουργηθεί από μία κοινότητα χαρτογράφων, οι οποίοι συνεισφέρουν και διατηρούν δεδομένα δρόμων, σιδηροδρομικών γραμμών, café, σιδηροδρομικών σταθμών και πολλά άλλα ανά τον κόσμο.

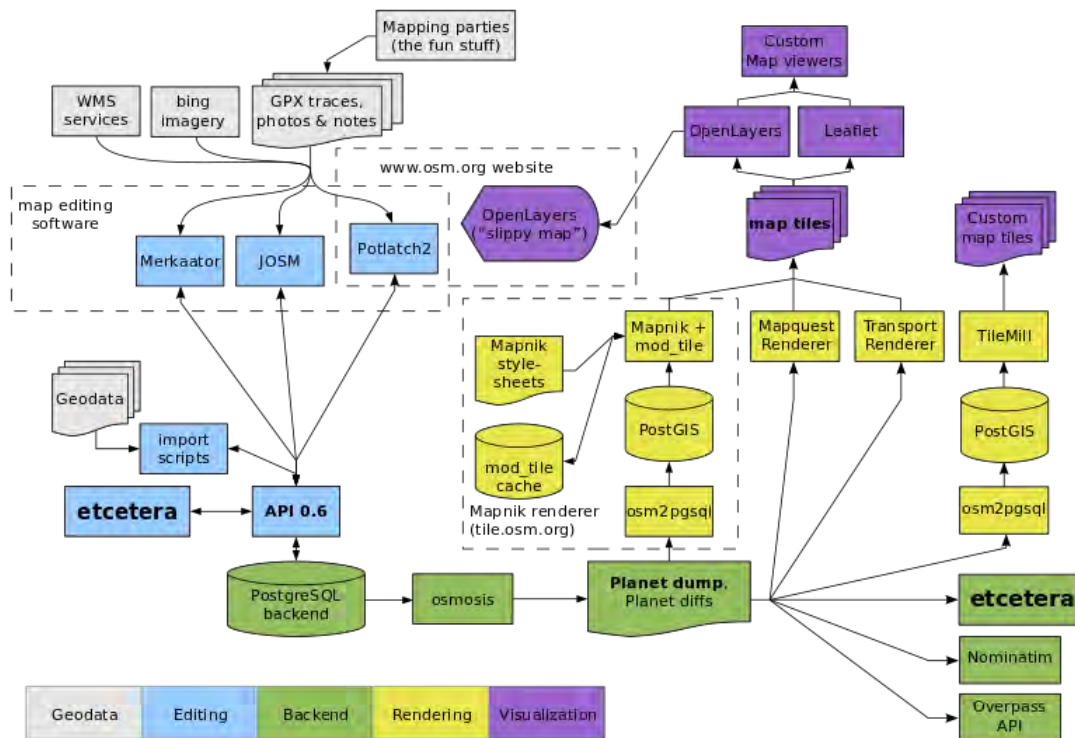
Η κοινότητα έχει μεγάλη ποικιλομορφία, είναι εξαιρετικά ενθουσιώδης και συνεχώς μεγαλώνει. Η ομάδα μας αποτελείται επιπλέον από επαγγελματίες γεωγραφικών πληροφοριακών συστημάτων, μηχανικούς που υποστηρίζουν τους OSM Servers, ανθρωπολόγους που χαρτογραφούν περιοχές που έχουν καταστραφεί και άλλους.

Η πρόσβαση είναι ελεύθερη με ανοικτά δεδομένα, που σημαίνει ότι οποιοσδήποτε επιθυμεί τα χρησιμοποιεί με οποιονδήποτε τρόπο, αρκεί να μνημονεύσει την προέλευσή τους, δηλαδή το OpenStreetMap και όσους συνεισφέρουν σε αυτό.



Εικόνα 1.2.10 Χάρτης OpenStreetMap

Η αρχιτεκτονική του χωρίζεται σε διακριτά κομμάτια που το καθένα εξελίσσεται σε συνδυασμό με τα υπόλοιπα. Απαρτίζεται από τα : Backend, Editing, Rendering, Geodata και Visualization όπως φαίνονται και στην παρακάτω εικόνα, όπου τα ξεχωρίζουμε και κατά χρώμα.



Εικόνα 1.2.11 Αρχιτεκτονική του OpenStreetMap

#### ❖ Database

Το Backend στηρίζεται στην βάση δεδομένων PostgreSQL. Διαχειρίζεται γεωγραφικά δεδομένα με την μορφή nodes, relations και ways.

#### ❖ API

Το OSM API είναι ένα REST Web Service που διευκολύνει το διάβασμα και το γράψιμο στην βάση δεδομένων. Η ανταλλαγή δεδομένων γίνεται με XML και με πρωτόκολλο επικοινωνίας HTTP. Η λογική του API στηρίζεται στο στο ίδιο Ruby on Rails που χρησιμοποιεί και εφαρμογή για το front end και το website.

#### ❖ OSM Front end

Το web front end είναι μια Ruby on Rails εφαρμογή, με views και models τοποθετημένες κοντά στις άλλες του XML API. Όπως συνηθίζεται σε κάθε Rails εφαρμογή, η σύνδεση με την βάση πραγματοποιείται απευθείας. Οι βασικές λειτουργίες του website είναι η προβολή του Slippy Map και του online editor iD. Επιπλέον όμως διαχειρίζεται και άλλες λειτουργίες όπως logging in, registering, GPS traces, data browser και άλλα.

##### ο Slippy Map [7]

Ο βασικός χάρτης που εμφανίζεται στην σελίδα του openstreetmap.org είναι ουσιαστικά ένα javascript interface το οποίο μας επιτρέπει να κάνουμε λειτουργίες zoom και pan. Τιν στιγμή που εκτελούνται αυτές οι λειτουργίες το website εφαρμόζει το Leaflet [8]. Ταυτόχρονα στέλνει τις συντεταγμένες



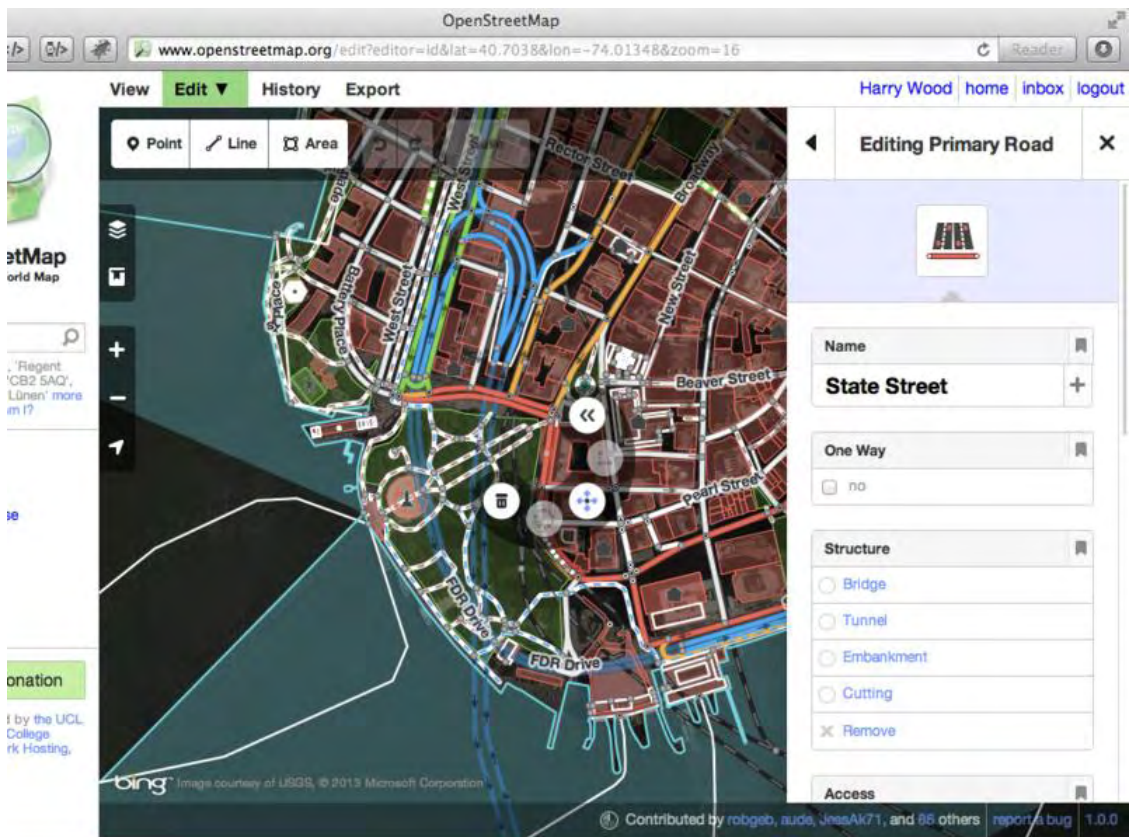
στον webserver και ανακαλεί τα κομμάτια μόνο του χάρτη που χρειάζονται και μας τα εμφανίζει.



Εικόνα 1.2.12 Slippy map

- iD [9]

Το iD είναι ένας online editor. Είναι γραμμένος με javascript και μπορεί να χρησιμοποιηθεί σε άλλα website για τα εμπλουτίσει με τις δυνατότητές του.



Εικόνα 1.2.13 iD Editor





### 1.3. Δομή Κεφαλαίων

Σε αυτό το σημείο, κρίνεται σκόπιμο να αναφερθούμε συνοπτικά στο περιεχόμενο των κεφαλαίων που ακολουθούν. Τα κεφάλαια που ακολουθούν είναι τα παρακάτω:

#### Κεφάλαιο 2 : Δομή πλατφόρμας

Στο κεφάλαιο αυτό θα παρουσιάσουμε την δομή και την αρχιτεκτονική της πλατφόρμας. Θα γίνει λεπτομερής αναφορά στην «Βάση Δεδομένων» και τον σχεδιασμό της. Θα αναλύσουμε την λειτουργία της «Ιστοσελίδας «και τον τρόπο με τον οποίο ο «Σχεδιασμός Γεωγραφικών Ερωτημάτων» τροφοδοτεί την «Υπηρεσία Διαδραστικών Χαρτών».

Ένα επιπλέον κομμάτι που θα περιγράψουμε στο δεύτερο κεφάλαιο είναι η «Υπηρεσία Γεωγραφικών Δεδομένων REST API». Είναι μια διεπαφή προγραμματισμού η οποία επιτρέπει εξωτερικές εφαρμογές να συνδέονται στην πλατφόρμα και να αποθηκεύουν ή να αντλούν δεδομένα από αυτή.

Συμπληρωματικά περιγράφεται και η δομή μιας εξωτερικής εφαρμογής η οποία αξιοποιεί την υπηρεσία REST και τροφοδοτεί με γεωχρονικά δεδομένα την βάση δεδομένων της πλατφόρμας.

#### Κεφάλαιο 3 : Εργαλεία Ανάπτυξης και Προγραμματισμού

Το κεφάλαιο τρία παρουσιάζει όλα τα εργαλεία που χρησιμοποιήθηκαν στην ανάπτυξη της πλατφόρμας. Δίνεται ιδιαίτερη έμφαση στις τεχνικές και τα χαρακτηριστικά των εργαλείων καθώς και ο σκοπός για τον οποίο τα επιλέξαμε.

Κυρίαρχο ρόλο στην διαδικασία ανάπτυξης έπαιξαν η βάση δεδομένων PostgreSQL και η γλώσσα προγραμματισμού Embarcadero Delphi. Επιπλέον παρουσιάζονται η επέκταση FMSoft uniGUI της Embarcadero Delphi, καθώς και η βιβλιοθήκες Sencha Ext Js και Sencha Touch, με τις οποίες υλοποιήθηκε το διαδικτυακό κομμάτι καθώς και ο απαραίτητος web server. Επιπλέον με την χρήση κατάλληλων χαρακτηριστικών της Embarcadero Delphi αναπτύχθηκε και το ανάλογο Rest API. Το απαραίτητο testing περιβάλλον και το debugging του Rest API το ανέλαβε το Insomnia Rest Client.

Οι χάρτες στους οποίους στηρίχθηκε η προβολή των γεωγραφικών ερωτημάτων είναι οι Google maps, με το εξαιρετικής ποιότητας API που διαθέτουν. Η πλατφόρμα εκτελεί τα ερωτήματα στην Postgres και με κατάλληλο αλγόριθμο διαμορφώνονται και προβάλλονται στην ιστοσελίδα.



Ιδιαίτερα χρήσιμο εργαλείο είναι και το NGINX (διαβάζεται engine-x), το οποίο είναι ένας δυνατός web server, ο οποίος στην πλατφόρμα μας χρησιμοποιήθηκε για δύο λόγους. Ο πρώτος είναι η υλοποίηση της επιθυμητής λειτουργία του πρωτοκόλλου Https και ο δεύτερο η λειτουργία Reverse Proxy που διαθέτει.

#### **Κεφάλαιο 4 : Παρουσίαση Χρήσης Πλατφόρμας**

Η λειτουργία των συστατικών εφαρμογών της πλατφόρμας περιγράφεται σε αυτό το κεφάλαιο. Ο αναγνώστης θα εξοικειωθεί με την χρήση και τις λειτουργίες της. Το πρώτο που παρουσιάζεται είναι η «Ιστοσελίδα» η οποία εξυπηρετεί τους χρήστες διαχειριστές και τους καθοδηγεί-πληροφορεί για τις υπηρεσίες της. Θα περιγράψουμε τον τρόπο που δημιουργείται ένας λογαριασμός χρήστη, πως ενεργοποιείται και πως εισέρχεται στο σύστημα.

Η βασική λειτουργία των διαχειριστών είναι η «Διαχείριση Ερωτημάτων», όπου έχει υλοποιηθεί μια κονσόλα διαχείρισης με πλούσια χαρακτηριστικά, με παράλληλη οπτικοποίηση του αποτελέσματος των γεωγραφικών ερωτημάτων. Γίνεται εκτενείς αναφορά στην λειτουργία της και περιγράφονται αναλυτικά οι εξελιγμένες δυνατότητές της.

Το αποτέλεσμα των γεωγραφικών ή γεωχρονικών ερωτημάτων των διαχειριστών, προσφέρεται στους τελικούς χρήστες της πλατφόρμας μέσω της «Υπηρεσίας Διαδραστικών Χαρτών». Αυτή η υπηρεσία παρουσιάζει με πολύ λειτουργικό τρόπο τα ερωτήματα και υποστηρίζει φορητές συσκευές και κινητά τηλέφωνα όλων των κατασκευαστών, με σύγχρονο φυλλομετρητή.

Θα παρουσιάσουμε επίσης την χρήση της «Υπηρεσίας Γεωγραφικών Δεδομένων Rest API» με την οποία θα περιγράψουμε ένα σύντομο αλλά ολοκληρωμένο παράδειγμα, το οποίο θα αναδείξει όλες τις δυνατότητες και λειτουργίες του.

Επιπλέον θα δείξουμε πως λειτουργεί η εφαρμογή που αναπτύχθηκε για φορητές συσκευές και κινητά, η οποία θα συλλέξει τα δεδομένα στιγμάτων και θα τα στείλει στην υπηρεσία REST.

#### **Κεφάλαιο 5 : Ανάπτυξη Πλατφόρμας**

Στο κεφάλαιο αυτό θα αναλύσουμε τις τεχνικές και τους αλγόριθμους που χρησιμοποιήθηκαν σε σημεία κλειδιά της πλατφόρμας. Θα αναλύσουμε και θα παρουσιάσουμε τον κώδικα που κρύβεται στην διαδικασία login, logout, θα εξηγήσουμε τον τρόπο με τον οποίο ενσωματώσαμε τους χάρτες Google και τους συνδυάσαμε με τις τεχνολογίες που χρησιμοποιήθηκαν. Επιπλέον θα δούμε την υλοποίηση του Rest API καθώς και της εφαρμογής για κινητά που το χρησιμοποιεί. Τέλος θα παρουσιάσουμε την παραμετροποίηση του NGINX που συντελεί στην τελική παρουσίαση της πλατφόρμας.



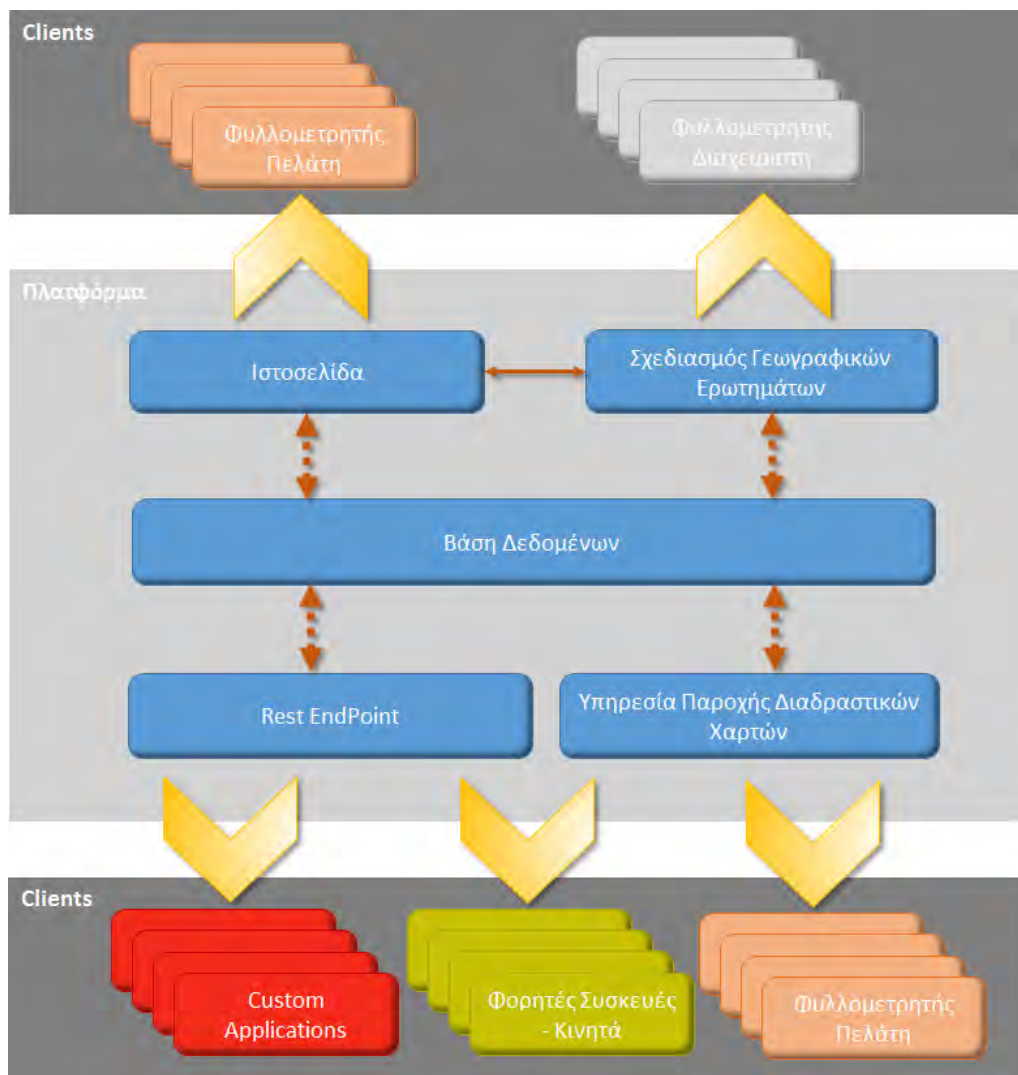
## **Κεφάλαιο 6 : Αξιολόγηση και Συμπεράσματα**

Στο τελευταίο κεφάλαιο θα κάνουμε την αξιολόγηση της μεταπτυχιακής διατριβής και θα επικεντρωθούμε σε μερικά κομβικά σημεία που συντελούν στην επιτυχία της πλατφόρμας. Θα ολοκληρώσουμε με κάποια συμπεράσματα και σκέψεις που θα μπορούσαν να χρησιμοποιηθούν μελλοντικά στην εξέλιξή της.

## 2. Δομή Πλατφόρμας

Η πλατφόρμα για την ανάπτυξη κινητών και διαδικτυακών γεωγραφικών εφαρμογών αποτελείται από ένα σύνολο βασικών υποσυστημάτων τα οποία εξυπηρετούνται από συγκεκριμένες εφαρμογές, όπως θα δούμε στο «Κεφάλαιο 5». Αυτά τα υποσυστήματα παρατίθενται στην παρακάτω λίστα και στην συνέχεια θα αναφερθούμε στο καθένα εκτενώς.

1. Βάση Δεδομένων
2. Ιστοσελίδα - Website
3. Σχεδιασμός Γεωγραφικών Ερωτημάτων
4. Υπηρεσία Γεωγραφικών Δεδομένων τύπου REST
5. Υπηρεσία Παροχής Διαδραστικών Χαρτών
6. Εφαρμογή Κινητού, Συλλογή Δεδομένων GPS

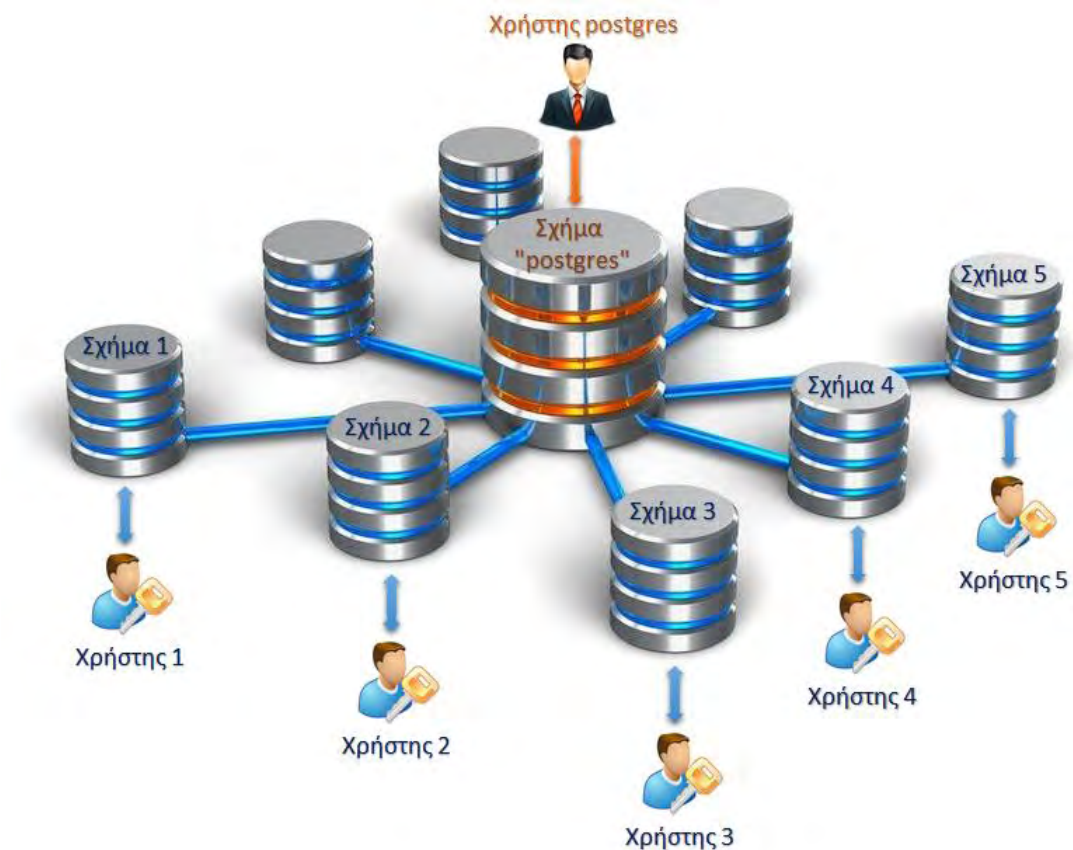


Εικόνα 2.1 Δομή της πλατφόρμας

## 2.1. Βάση Δεδομένων

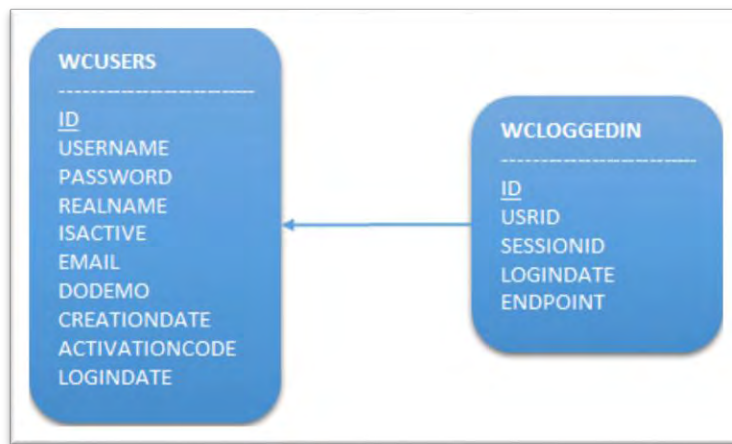
Η αρχιτεκτονική της βάσης δεδομένων ακολουθεί την εξής φιλοσοφία. Υπάρχει ένα schema με όνομα “postgres” στην βάση δεδομένων το οποίο αποθηκεύει τα στοιχεία των χρηστών-διαχειριστών του γεωγραφικού μας συστήματος.

Για κάθε ένα διαχειριστή η εφαρμογή δημιουργεί νέο, ανεξάρτητο σχήμα βάσης δεδομένων, όπου ο διαχειριστής θα μπορεί να δημιουργήσει πίνακες με γεωγραφικό και γεωχρονικό περιεχόμενο και να τους ενημερώσει. Έχει δοθεί μεγάλη σημασία στην αρχιτεκτονική των σχημάτων ούτως ώστε να εξασφαλίζεται η απομόνωση των δεδομένων με στόχο την μεγιστοποίηση της ασφάλειας. Επιπλέον στο σχήμα του διαχειριστή αποθηκεύονται τα γεωγραφικά ερωτήματα, και οι παράμετροί τους.



Εικόνα 2.1.1 Βάση δεδομένων πλατφόρμας, Σχήμα “postgres” και Σχήματα χρηστών

Το schema “postgres” αποτελείται από δύο πίνακες, ένας πίνακας περιέχει τα στοιχεία χρήστη και ο δεύτερος πληροφορίες σύνδεσης και έχει την παρακάτω μορφή :



Εικόνα 2.1.2 Σχήμα “postgres”

#### Πίνακας WCUSERS

Είναι ο πίνακας όπου αποθηκεύουμε τα στοιχεία των χρηστών. Τα πεδία του πίνακα και ο συνοπτικός τους σχολιασμός φαίνονται παρακάτω:

Πεδίο	Τύπος	Σχόλια
ID	Ακέραιος	Πρωτεύον κλειδί πίνακα
USERNAME	Κείμενο	Όνομα χρήστη (nickname)
PASSWORD	Κείμενο	Κωδικοποιημένο με Digest 5 κλειδί χρήστη
REALNAME	Κείμενο	Ονοματεπώνυμο Χρήστη
ISACTIVE	Μικρός Ακέραιος	Έχει την τιμή “1” για ενεργούς χρήστες, “0” για ανενεργούς
EMAIL	Κείμενο	Το email του χρήστη
DODEMO	Μικρός Ακέραιος	Αν έχει την τιμή “1” ο χρήστης ζήτησε να δημιουργηθούν δεδομένα επίδειξης
CREATIONDATE	Ημερομηνία	Ημερομηνία δημιουργίας Λογαριασμού
ACTIVATIONCODE	Κείμενο	Κωδικός ενεργοποίησης, πιστοποίηση ορθότητας email
LOGINDATE	Ημερομηνία	Ημερομηνία τελευταίας εισαγωγής στο σύστημα

#### Πίνακας WCLOGGEDIN

Σε αυτόν το πίνακα αποθηκεύουμε τις επιτυχημένες εισόδους (login) στο σύστημα. Επιπλέον αποθηκεύουμε και το σημείο εισόδου, δηλαδή από ποια εφαρμογή έγινε το login.

Πεδίο	Τύπος	Σχόλια
ID	Ακέραιος	Πρωτεύον κλειδί πίνακα
USRID	Ακέραιος	ID χρήστη, foreign key στο πεδίο ID το WCUSERS
SESSIONID	Κείμενο	Μοναδικό ID που αποδίδεται σε κάθε νέο session
LOGINDATE	Ημερομηνία	Ημερομηνία εισαγωγής του session
ENDPOINT	Κείμενο	Σημείο εισαγωγής

Εκτός από το βασικό schema “postgres”, για κάθε χρήστη-διαχειριστή δημιουργείται ένα αυτόνομο, απομονωμένο και ανεξάρτητο schema για να φιλοξενήσει τα γεωγραφικά και γεωχρονικά δεδομένα του. Ο διαχειριστής έχει πλήρη δικαιώματα στο schema του και μπορεί



να δημιουργήσει όσους και οποιασδήποτε μορφής πίνακες επιθυμεί. Σε κάθε νέο schema δημιουργούνται δύο πίνακες, όπου αποθηκεύονται στον ένα τα γεωγραφικά ερωτήματα και στον άλλο οι παράμετροι των ερωτημάτων. Το schema που δημιουργείται για κάθε νέο χρήστη έχει την παρακάτω μορφή :



Εικόνα 2.1.3 Σχήμα Χρήστη

#### Πίνακας WCQUERIES

Είναι ένας πίνακας όπου αποθηκεύουμε στο σχήμα του χρήστη, τα γεωγραφικά μας ερωτήματα. Παρακάτω εμφανίζονται τα πεδία του με συνοπτικά σχόλια:

Πεδίο	Τύπος	Σχόλια
ID	Ακέραιος	Πρωτεύον κλειδί πίνακα
USRID	Ακέραιος	ID χρήστη
CAPTION	Κείμενο	Τίτλος Ερωτήματος
COLOR	Κείμενο	Χρώμα με το οποίο το ερώτημα θα εμφανισθεί
SQL	Κείμενο	Περιέχει το SQL Query
GEOMCOLUMN	Κείμενο	Όνομα στήλης γεωγραφικών δεδομένων
ISVISIBLE	Μικρός Ακέραιος	Αν έχει την τιμή “1” ο ερώτημα εμφανίζεται, “0” διαφορετικά
LINENUM	Ακέραιος	Αύξων αριθμός ερωτήματος
DESCRCOLUMN	Κείμενο	Στήλη με περιεχόμενο κειμένου πληροφορίας

#### Πίνακας WCPARAMS

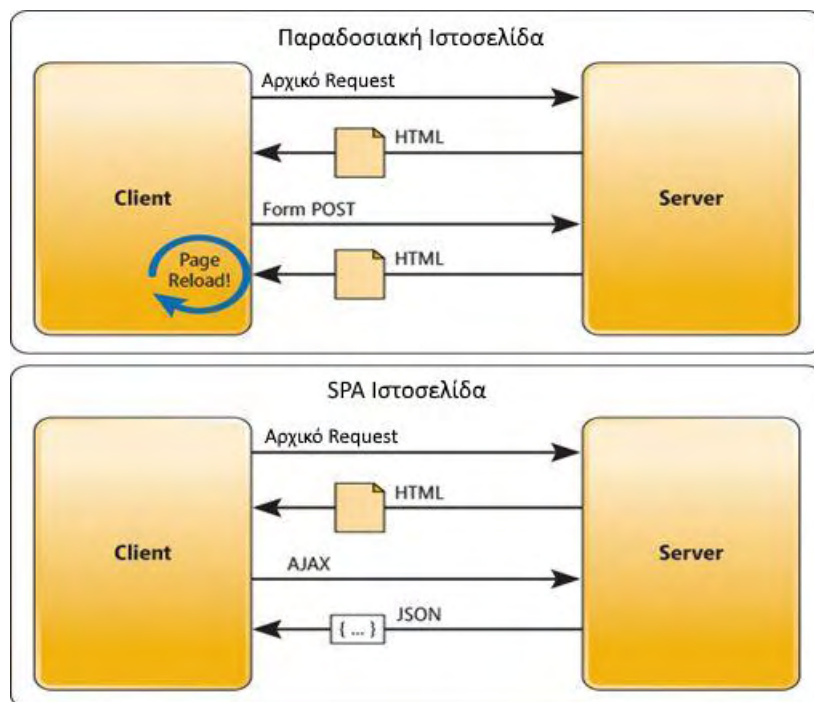
Είναι ένας πίνακας όπου αποθηκεύουμε παραμέτρους χρήστη, για χρήση στα γεωγραφικά μας ερωτήματα.

Πεδίο	Τύπος	Σχόλια
ID	Ακέραιος	Πρωτεύον κλειδί πίνακα
PARAMNAME	Κείμενο	Όνομα παραμέτρου
DATATYPE	Κείμενο	Τύπος δεδομένων παραμέτρου
WHEREPART	Κείμενο	Περιέχει μέρος από το SQL Query
CAPTION	Κείμενο	Τίτλος παραμέτρου
VALUESQUERY	Κείμενο	SQL Query για παράμετρο, τύπου λίστας
DEFAULTVALUE	Κείμενο	Προεπιλεγμένη τιμή παραμέτρου

MINVALUE	Κείμενο	Ελάχιστη τιμή παραμέτρου
MAXVALUE	Κείμενο	Μέγιστη τιμή παραμέτρου

## 2.2. Ιστοσελίδα

Η ιστοσελίδα της εφαρμογής περιέχει το λειτουργικό κομμάτι της διεπαφής του χρήστη. Η αρχιτεκτονική της ιστοσελίδας βασίζεται στην μοντέλο Single-page application [10] (SPA). Σε αντίθεση με της παραδοσιακές ιστοσελίδες στις οποίες υπάρχει συχνή φόρτωση σελίδων, στις SPA ιστοσελίδες η σελίδα φορτώνεται μια φορά και η παρουσίαση του περιεχομένου της γίνεται δυναμικά, με το πρόγραμμα που φορτώθηκε σε αυτή.



Εικόνα 2.2.1 Σύγκριση παραδοσιακών ιστοσελίδων με SPA ιστοσελίδες

Όσον αφορά το περιεχόμενο της ιστοσελίδας, η ιστοσελίδα μας αποτελείται από τις εξής σελίδες :

1. Κεντρική Σελίδα

Περιέχει κείμενα που καθοδηγούν ένα νέο επισκέπτη και εξηγεί με λίγα λόγια την γενικότερη λειτουργία του συστήματος.

2. Σελίδα αίτησης νέου λογαριασμού χρήστη

Η σελίδα ζητά ορισμένες βασικές πληροφορίες προκειμένου να δημιουργήσει ένα νέο λογαριασμό.

3. Σελίδα επιβεβαίωσης στοιχείων χρήστη

Επιβεβαιώνονται τα στοιχεία του χρήστη, με την εισαγωγή κωδικού ενεργοποίησης που απεστάλη με email.





4. Σελίδα Πληροφοριών

Περιέχει αναλυτικές πληροφορίες σχετικά με την λειτουργία της εφαρμογής

5. Σελίδα Διαχείρισης Γεωγραφικών Ερωτημάτων

Κατευθύνει τον χρήστη στην εφαρμογή Διαχείρισης Γεωγραφικών Ερωτημάτων.

### 2.3. Σχεδιασμός Γεωγραφικών Ερωτημάτων

Το υποσύστημα του Σχεδιασμού Γεωγραφικών Ερωτημάτων έχει σαν σκοπό την δημιουργία και διαχείριση των γεωγραφικών ερωτημάτων που επιθυμεί ο διαχειριστής να παρουσιάσει σε μια σελίδα, σε μια εφαρμογή ή σε μια φορητή συσκευή. Του δίνεται η δυνατότητα να εισάγει παραμέτρους στα γεωγραφικά του ερωτήματα προκειμένου να δημιουργηθεί η διεπαφή, με την απαραίτητη διαδραστικότητα, με τον τελικό χρήστη.

Στην σελίδα αυτή, ο διαχειριστής έχει στην διάθεσή του, εκτός από τα προσφερόμενα εργαλεία, ένα χάρτη Google στον οποίο θα οπτικοποιεί τα ερωτήματά του. Στο σχεδιασμό των γεωγραφικών ερωτημάτων συνδυάζονται τα δεδομένα του σχήματος διαχειριστή με την δυναμική προβολή των Google maps.

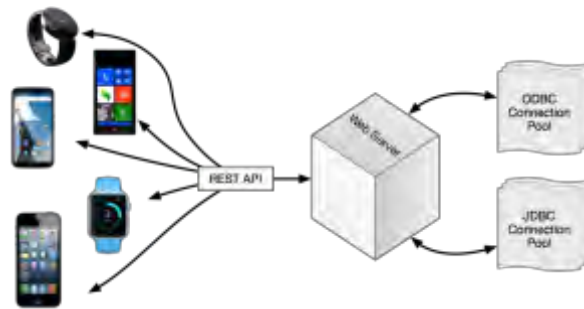
### 2.4. Υπηρεσία Διαδραστικών Χαρτών

Η υπηρεσία αυτή αναπτύχθηκε με σκοπό να προσφέρει η εφαρμογή μας ένα εύκολο και αποτελεσματικό τρόπο παρουσίασης των διαδραστικών γεωγραφικών χαρτών στην συσκευή του τελικού χρήστη. Λαμβάνει τα γεωγραφικά δεδομένα από τα προσχεδιασμένα ερωτήματα του διαχειριστή και τα προβάλλει πάνω στον χάρτη Google. Επιπλέον εμφανίζει τις παραμέτρους του ερωτήματος και επιτρέπει στον τελικό χρήστη να τις μεταβάλλει, προκειμένου να εμφανίσει το αποτέλεσμα που επιθυμεί.

Μια χαρακτηριστική ιδιότητα των διαδραστικών χαρτών είναι ότι ο χρήστης-διαχειριστής μπορεί με την δημιουργία κατάλληλου σχήματος να σχεδιάσει και γεωχρονικά ερωτήματα. Στο παράδειγμα της εφαρμογής συλλογής δεδομένων, που θα παρουσιασθεί παρακάτω, γίνεται χρήση γεωχρονικών δεδομένων.

### 2.5. Υπηρεσία Γεωγραφικών Δεδομένων REST API

Η υπηρεσία Rest [11] (Representational state transfer) υλοποιεί την απαραίτητη διεπαφή με άλλα προγράμματα, τα οποία θα συνδεθούν στην υποδομή. Έχουν υλοποιηθεί όλες οι βασικές λειτουργίες όπως σύνδεση, αποσύνδεση με κωδικό χρήστη, εισαγωγή και εξαγωγή δεδομένων.



Εικόνα 2.5 Επικοινωνία με το Rest API

Η απαραίτητη επικοινωνία με την υπηρεσία REST έχει σαν βάση το πρότυπο JSON [12]. Το πρότυπο αυτό είναι το πλέον γενικό πρότυπο ανταλλαγής δεδομένων. Με αυτό τον τρόπο θα επιτευχθεί και η απαραίτητη επικοινωνία με οποιαδήποτε εξωτερική εφαρμογή.

Το μεγαλύτερο προτέρημα του REST είναι ότι είναι stateless πρωτόκολλο το οποίο δεν επηρεάζεται από απώλειες συνδέσεων, σε αντίθεση με τα ODBC, JDBC και OLE DB. Οι κλήσεις προς αυτό υλοποιούνται ως URIs, τα οποία λαμβάνει ο web server, τα μετατρέπει σε SQL statements και τα προωθεί προς την βάση δεδομένων.

Για τις ανάγκες διασύνδεσης της πλατφόρμας με εξωτερικές εφαρμογές η υλοποίηση της υπηρεσίας περιορίσθηκε στα απολύτως αναγκαία, διατηρώντας όμως την ανοικτή της αρχιτεκτονική. Στο κεφάλαιο 6 θα προτείνουμε τα απαραίτητα συμπληρωματικά στοιχεία που θα συνθέσουν μια πλήρως ανεπτυγμένη Rest υπηρεσία.

## 2.6. Εφαρμογή Κινητού, Συλλογή Δεδομένων GPS

Ένα απαραίτητο συμπλήρωμα για την ολοκλήρωση της μεταπτυχιακής μου διατριβής είναι και η εφαρμογή για την συλλογή δεδομένων GPS. Αυτή η εφαρμογή εκτελείται σε κινητά και επικοινωνεί με την υπηρεσία REST. Η λειτουργία της είναι αποκλειστικά η αποστολή του στίγματος και της ημερομηνίας και ώρας του κινητού στον απομακρυσμένο server. Με αυτό τον τρόπο μπορούμε να εκμεταλλευτούμε τα παραγόμενα γεωχρονικά δεδομένα και να τα παρουσιάσουμε με την υπηρεσία διαδραστικών χαρτών ή με μια άλλη εξωτερική εφαρμογή η οποία συνδέεται στην υπηρεσία REST.



*Βελέντζας Πολυχρόνης*  
*Διαχείριση Γεωγραφικών Ερωτημάτων και*  
*Υπηρεσία Διαδραστικών Χαρτών για Ιστοσελίδες*

## 3.Εργαλεία Ανάπτυξης και Προγραμματισμού

Πριν προχωρήσουμε στην περιγραφή της ανάπτυξης της εφαρμογής κρίνεται απαραίτητο να αναφερθούμε στα εργαλεία και τις τεχνολογίες που χρησιμοποιήθηκαν. Η μεταπτυχιακή διατριβή « Διαχείριση Γεωγραφικών Ερωτημάτων και Υπηρεσία Διαδραστικών Χαρτών για Ιστοσελίδες» υλοποιήθηκε με την βοήθεια των παρακάτω :

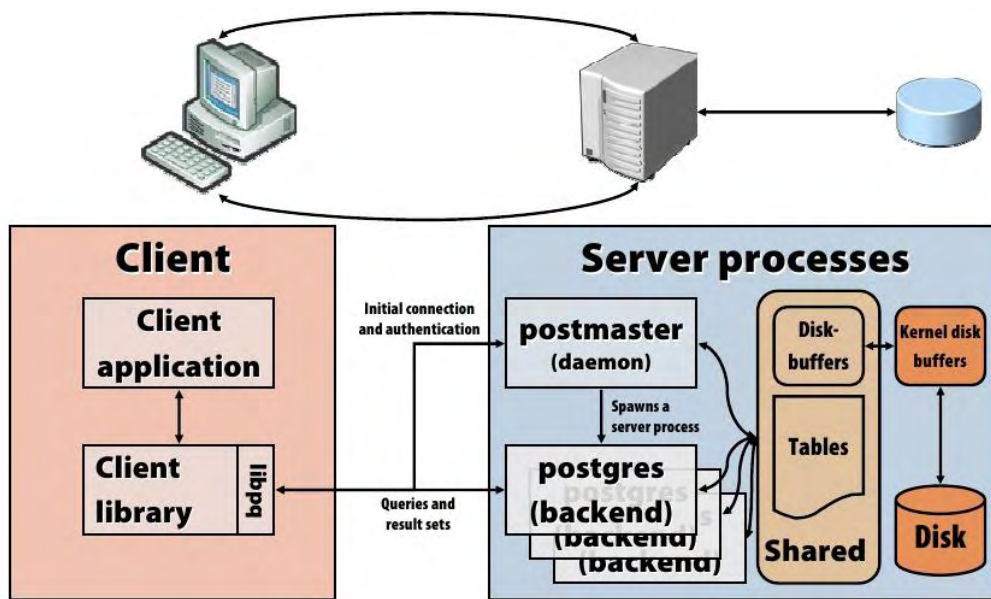
1. PostgreSQL 9.5, PostGIS 2.3 Extension
2. Embarcadero Delphi XE 10.2
3. FMSoft Unigui 1.0.0.1373
4. Sencha Ext JS, Sencha Touch
5. NGINX 1.11.11
6. Google Map API v3
7. Insomnia Rest Client v5

### 3.1. PostgreSQL 9.5, PostGIS 2.3 Extension

Η PostgreSQL [13] [14], ή πιο απλά Postgres, είναι μια Object Relational Database (ORDBMS). Δηλαδή είναι μια σχεσιακή βάση δεδομένων με επιπλέον κάποια χαρακτηριστικά αντικειμένων. Στην Postgres δίνεται ιδιαίτερη έμφαση στην επεκτασιμότητα και στην ορθή συμμόρφωση με τα στάνταρντ. Είναι ένας database server και η πρωταρχική του λειτουργία είναι να αποθηκεύει με ασφάλεια τα δεδομένα και τα επιστρέφει εξυπηρετώντας κλήσεις από άλλες εφαρμογές. Μπορεί να εξυπηρετήσει φόρτο εργασίας από μικρές εφαρμογές που τρέχουν σε ένα υπολογιστή έως μεγάλες εφαρμογές, προσανατολισμένες στο διαδίκτυο, με πάρα πολλούς ταυτόχρονους χρήστες. Υποστηρίζει όλα τα γνωστά λειτουργικά συστήματα.

Η Postgres συμμορφώνεται με το πρότυπο ACID (Atomicity, Consistency, Isolation, Durability) και υποστηρίζει συναλλαγές (transactional). Εν συντομία η Postgres υποστηρίζει τα updatable views, materialized views, triggers, foreign keys, καθώς επίσης τα functions, stored procedures και άλλες επεκτάσεις.

Η Postgres αναπτύσσεται από την PostgreSQL Global Development Group, μια σύνθετη ομάδα από πολλές εταιρείες και αυτόνομους προγραμματιστές. Είναι ελεύθερο και ανοιχτού κώδικα και διανέμεται με τους όρους της PostgreSQL άδειας.



Εικόνα 3.1.1 Αρχιτεκτονική PostgreSQL

Για τις ανάγκες των εφαρμογών που αναπτύχθηκαν, η PostgreSQL επεκτάθηκε με την προσθήκη PostGIS [15] [16]. Το PostGIS είναι λογισμικό ανοικτού κώδικα και προσθέτει γεωγραφικά αντικείμενα στην PostgreSQL. Το PostGIS ακολουθεί τις SQL προδιαγραφές του Open GeoSpatial Consortium (OGC). Ως αποτέλεσμα η PostgreSQL εμπλουτίζεται με τα ακόλουθα χαρακτηριστικά :

- ❖ Geometry types for Points, LineStrings, Polygons, MultiPoints, MultiLineStrings, Multipolygons and GeometryCollections.
- ❖ Spatial predicates for determining the interactions of geometries using the 3x3 DE-9IM (provided by the GEOS software library).
- ❖ Spatial operators for determining geospatial measurements like area, distance, length and perimeter.
- ❖ Spatial operators for determining geospatial set operations, like union, difference, symmetric difference and buffers (provided by GEOS).
- ❖ R-tree-over-GiST (Generalized Search Tree) spatial indexes for high speed spatial querying.
- ❖ Index selectivity support, to provide high performance query plans for mixed spatial/non-spatial queries.
- ❖ For raster data, PostGIS WKT Raster (now integrated into PostGIS 2.0+ and renamed PostGIS Raster)

### 3.2. Embarcadero Delphi XE 10.2

Η Embarcadero Delphi [17] [18], είναι μια από τις πιο διαχρονικές γλώσσες προγραμματισμού, μιας και έχει βασιστεί στην διάλεκτο Object Pascal. Η πρώτη Delphi [19] (Delphi 1) ξεκίνησε το 1995 για τα 16bit Windows 3.1. Ήταν ένα πολύ εξελιγμένο εργαλείο για την εποχή εκείνη και μας προϋδέασε για τα Rapid Application Development (RAD) εργαλεία που ακολούθησαν.

Μετά από πολλά χρόνια και πάρα πολλές εξελίξεις, αυτό το ώριμο εργαλείο έχει εκσυγχρονισθεί και υποστηρίζει όλες τις νέες τεχνολογίες. Τα δυνατά του σημεία ήταν εξ αρχής το πολύ γρήγορο compilation και το ότι το παραγόμενο εκτελέσιμο δεν είχε εξαρτήσεις από άλλες βιβλιοθήκες. Τα τελευταία χρόνια του πρόσθεσαν την δυνατότητα να παράγει κώδικα και για άλλα λειτουργικά συστήματα, για φορητές συσκευές, κινητά τηλέφωνα και γενικότερα για Internet of Things (IoT).



Εικόνα 3.2.1 Υποστηριζόμενα Λειτουργικά Συστήματα Delphi

Για κάθε ένα από τα υποστηριζόμενα λειτουργικά συστήματα (Windows, OSX, Android, Linux) [20], η Delphi έχει από ένα compiler ο οποίος παράγει πραγματικά native Intel ή ARM v7 binary εκτελέσιμα αρχεία. Οι παραγόμενες native εφαρμογές, είναι βελτιστοποιημένες για κάθε πλατφόρμα υλικού, με αποτέλεσμα το πλουσιότερο και γρηγορότερο τελικό αποτέλεσμα. Αναλυτικότερα, τα προτερήματα των native εφαρμογών Delphi είναι τα παρακάτω :

- ❖ **Ταχύτητα**

Χωρίς περιορισμούς, απόλυτη εκμετάλλευση του υλικού και της CPU, χωρίς τα αρνητικά αποτελέσματα των script interpreters και των VM.

- ❖ **Έμφυτη ασφάλεια**

Πολλοί προγραμματιστές χρησιμοποιούν Java, Javascript και διάφορα WebKits τα οποία παρουσιάζουν κενά ασφαλείας. Ο native κώδικας της Delphi δεν στηρίζεται σε τέτοιες βιβλιοθήκες και οι φορητές συσκευές δεν κινδυνεύουν

❖ **Προβλεψιμότητα**

Οι εφαρμογές τρέχουν απευθείας στην CPU, όπως θέλουν πραγματικά οι κατασκευαστές συσκευών και δεν καθυστερούν από επιπλέον επίπεδα εφαρμογών και από τεχνικές τυχαίου garbage collection. Στις φορητές συσκευές ο compiler χρησιμοποιεί ένα εξελιγμένο και ντετερμινιστικό Automatic Reference Counter (ARC) διαχειριστή μνήμης.

❖ **Καλύτερη εμπειρία τελικού χρήστη**

Οι εφαρμογές εκμεταλλεύονται πλήρως το υλικό και τις ικανότητες των συσκευών, όπως οι αισθητήρες και η κάμερα και επιπλέον μπορεί συνδεθεί με βιβλιοθήκες native και τρίτων κατασκευαστών.

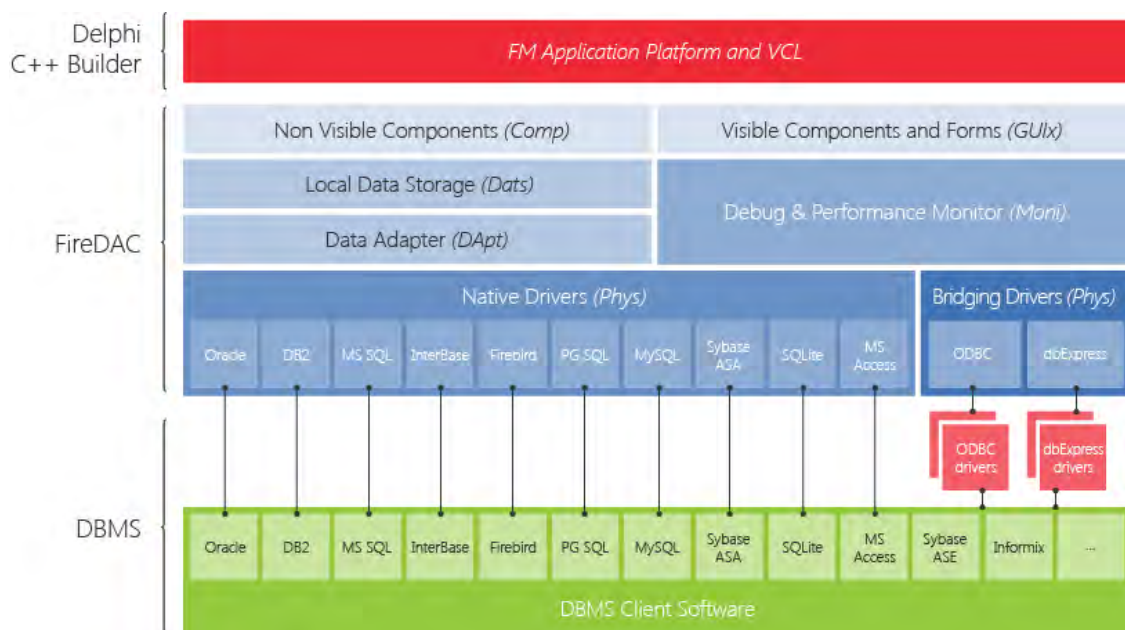
Οι δυνατότητες και τα χαρακτηριστικά της γλώσσας προγραμματισμού Delphi είναι πολλά. Μερικά από τα σημαντικότερα είναι τα παρακάτω:

❖ **Restful Web Services**

Υποστήριξη για Restful Web Services τόσο σε producer υπηρεσίες όσο και σε consumer εφαρμογές για desktop ή mobile πλατφόρμες.

❖ **Βελτιστοποιημένη Πρόσβαση σε Βάσεις Δεδομένων**

Υπάρχουν αρκετές βιβλιοθήκες-τεχνικές για πρόσβαση σε βάσεις δεδομένων όπως dbExpress, dbGo και FireDAC [21]. Το FireDAC είναι η πιο πρόσφατη και πιο πολλά υποσχόμενη τεχνολογία, η οποία παρέχει απρόσκοπτη και διαφανή συνδεσιμότητα σε όλες τις βάσεις δεδομένων και σε όλα τα υποστηριζόμενα λειτουργικά συστήματα.

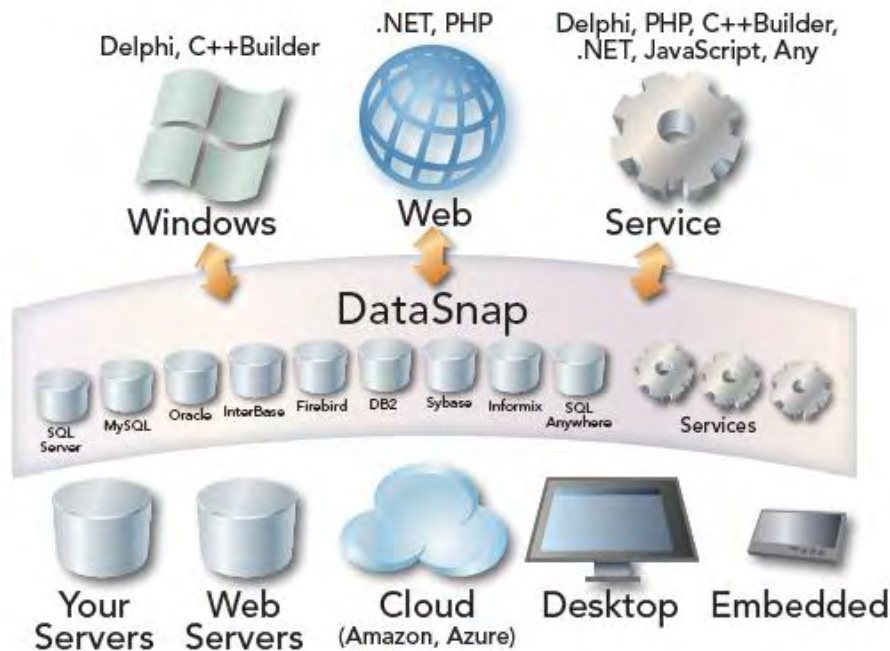


Εικόνα 3.2.2 Συνδεσιμότητα FireDAC



❖ **Datasnap**

Παλαιότερα γνωστό ως MIDAS, το datasnap είναι η τεχνολογία που επιτρέπει την δημιουργία multi-tier εφαρμογές βάσεων δεδομένων. Με τα datasnap το Delphi δημιουργεί client εφαρμογές που συνδέονται με ένα tier διαμέσου TCP/IP, DCOM, HTTP ή ακόμη SOAP πρωτόκολλο μεταφοράς δεδομένων.



Εικόνα 3.2.3 Datasnap

❖ **Indy (internet direct) [22]**

Το Indy είναι βιβλιοθήκη που περιλαμβάνει όλα τα απαραίτητα πρωτόκολλα επικοινωνίας του διαδικτύου. Πιο συγκεκριμένα έχει συστατικά client και server για TCP,UDP και Raw Sockets καθώς επίσης και για περισσότερα από 100 υψηλότερου επιπέδου πρωτόκολλα όπως SMTP, POP3, NNTP και HTTP.

### 3.3. FMSoft Unigui

Το uniGui [23] είναι ένα Web Application Framework το οποίο λειτουργεί μέσα από την γλώσσα προγραμματισμού Delphi. Επιτρέπει στους προγραμματιστές να δημιουργήσουν, να σχεδιάσουν και να αποσφαλματώσουν web εφαρμογές μέσα στο IDE της Delphi χρησιμοποιώντας ένα μοναδικό, πλήρες σετ συστατικών (visual components). Κάθε συστατικό έχει σχεδιαστεί να συμπεριφέρεται όμοια με το αντίστοιχο συστατικό του Delphi VCL (βασική βιβλιοθήκη της Delphi που χρησιμοποιείται για ανάπτυξη εφαρμογών για Windows). Ως



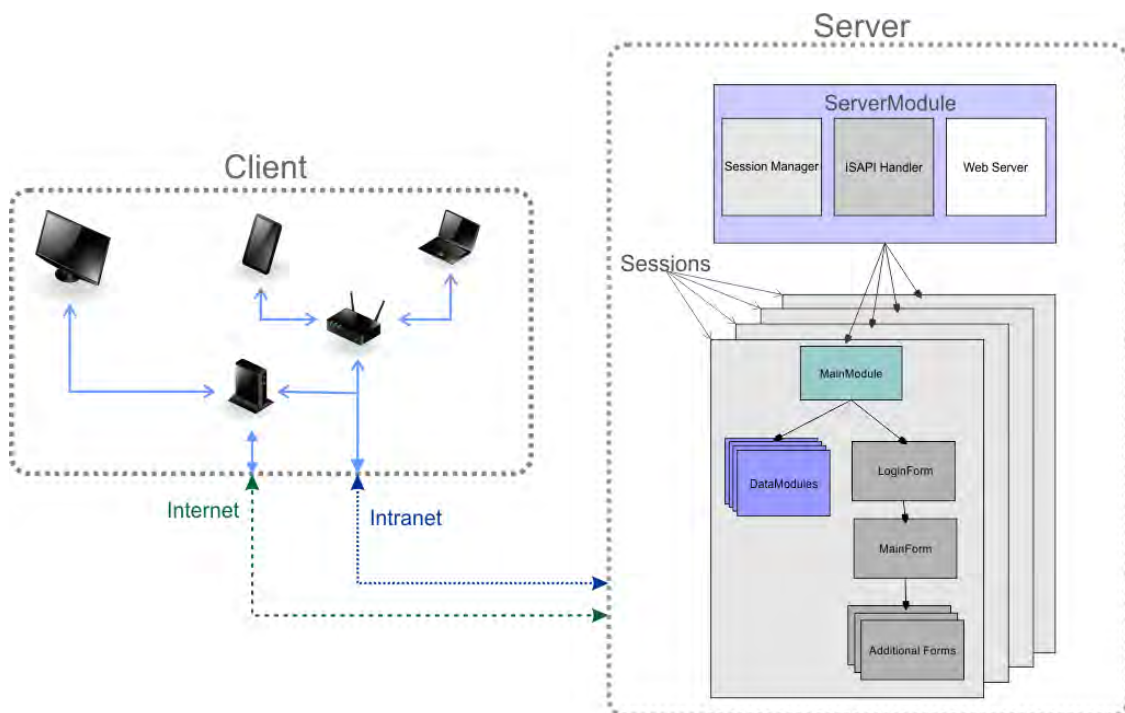
αποτέλεσμα αυτής της προσέγγισης, ο προγραμματιστής έχει μια πολύ εύκολη μετάβαση στο περιβάλλον του uniGui και χρησιμοποιεί την πρότερη γνώση του για να αναπτύξει πλέον web εφαρμογές.

Το uniGui βοηθά τους προγραμματιστές να επικεντρωθούν στην λογική της εφαρμογής και όχι στις λεπτομέρειες μιας web εφαρμογής όπως η απευθείας επεξεργασία αρχείων HTML, Javascript, XML templates, CSS και JSON. Με αυτόν τον τρόπο επιτυγχάνεται εξαιρετικό αποτέλεσμα σε πολύ λιγότερο χρόνο.

Ένα άλλο σημαντικό θέμα είναι που και με ποιόν τρόπο εγκαθίσταται η παραγόμενη εφαρμογή και πως συνδυάζεται με το οικοσύστημα των web servers. Ευτυχώς η λύση του uniGUI είναι ξεκάθαρη. Ο προγραμματιστής μπορεί να επιλέξει ένα τρόπο από τους :

- ❖ Υπηρεσία Windows (αναμένεται και σε Linux)
- ❖ Αυτόνομος Server (εκτελέσιμη εφαρμογή που είναι web server)
- ❖ ISAPI Module [24] (Συνδυάζεται με Microsoft IIS, Apache Web Server for Windows)

Το παραγόμενο αποτέλεσμα είναι εξαιρετικά ποιοτικό στον φυλλομετρητή του τελικού χρήστη. Εκτός από την αποτελεσματικότητα του Server το uniGUI έχει φροντίσει να παρέχει ένα εξίσου αποτελεσματικό περιβάλλον στην πλευρά του χρήστη με την βοήθεια του πακέτου Sencha Ext JS και Sencha Touch, το οποίο περιγράφεται παρακάτω.



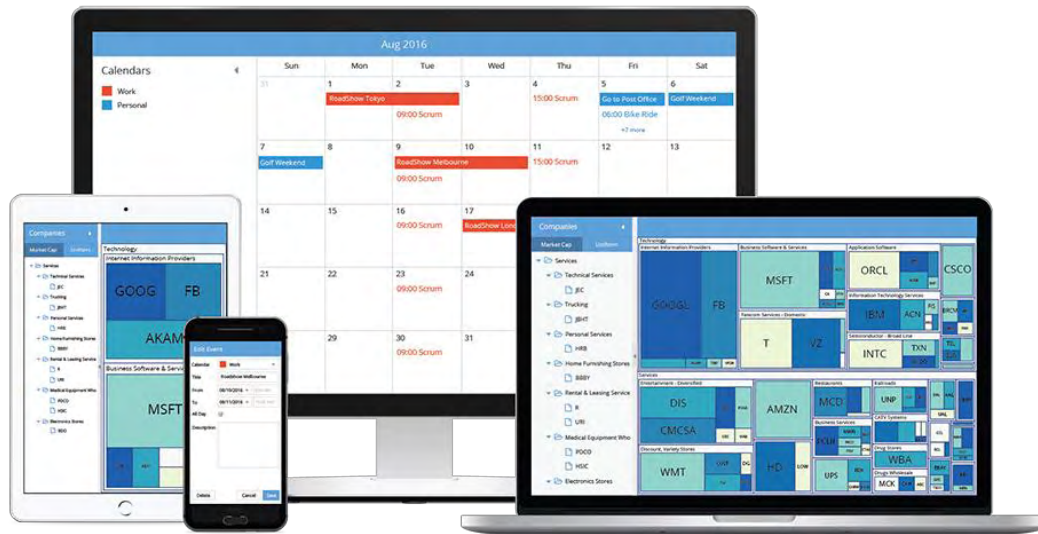
Εικόνα 3.3.1 Αρχιτεκτονική uniGUI

### 3.4. Sencha Ext Js, Sencha Touch

Το Sencha Ext Js [25] είναι ένα καθαρό Javascript application framework με το οποίο μπορεί κανείς να κατασκευάσει cross platform εφαρμογές, κάνοντας χρήση τεχνικών όπως Ajax, DHTML, HTML5 και DOM scripting. Το Sencha Ext Js δεν έχει καμία εξωτερική εξάρτηση από βιβλιοθήκες τρίτων κατασκευαστών.

Τα κυριότερα συστατικά του Sencha Ext Js είναι τα παρακάτω :

- ❖ **Text Field, Textarea**  
Συστατικά για εισαγωγή κειμένου
- ❖ **Date field**  
Επιλογή ημερομηνίας από αναδυόμενο (popup) ημερολόγιο
- ❖ **Numeric field**  
Πεδίο για εισαγωγή μόνο αριθμών
- ❖ **List box, Combo Box**  
Λίστες επιλογής από πίνακα τιμών
- ❖ **Radio, checkbox**  
Επιλογή με τσεκάρισμα
- ❖ **HTML Editor**  
Εμπλουτισμένος κειμενογράφος με δυνατότητες HTML
- ❖ **Grid Control**  
Συστατικό τύπου πίνακα, με δυνατότητες εισαγωγής, ενημέρωσης και διαγραφής. Επίσης παρέχει δυνατότητα σελιδοποίησης και ανάγνωσης από διαφορετικές πηγές.
- ❖ **Tree Control**  
Συστατικό τύπου δένδρου. Δημιουργεί πολλά ιεραρχικά επίπεδα με συνδεδεμένους κόμβους
- ❖ **Tab panels**  
Δημιουργία πολλαπλών επικαλυπτόμενων πάνελ με δομή tab.
- ❖ **Toolbar**  
Εργαλειοθήκες με εικονίδια
- ❖ **Menu**  
Μενού επιλογών όπως στις εφαρμογές επιτραπέζιων υπολογιστών.
- ❖ **Region Panels**  
Πάνελ περιοχών που επιτρέπουν τον διαχωρισμό μιας φόρμας σε πολλές υπο-περιοχές.
- ❖ **Sliders**  
Γραμμές κύλισης
- ❖ **Vector graphics charts**  
Διαγράμματα με δυνατότητες διανυσματικών γραφικών

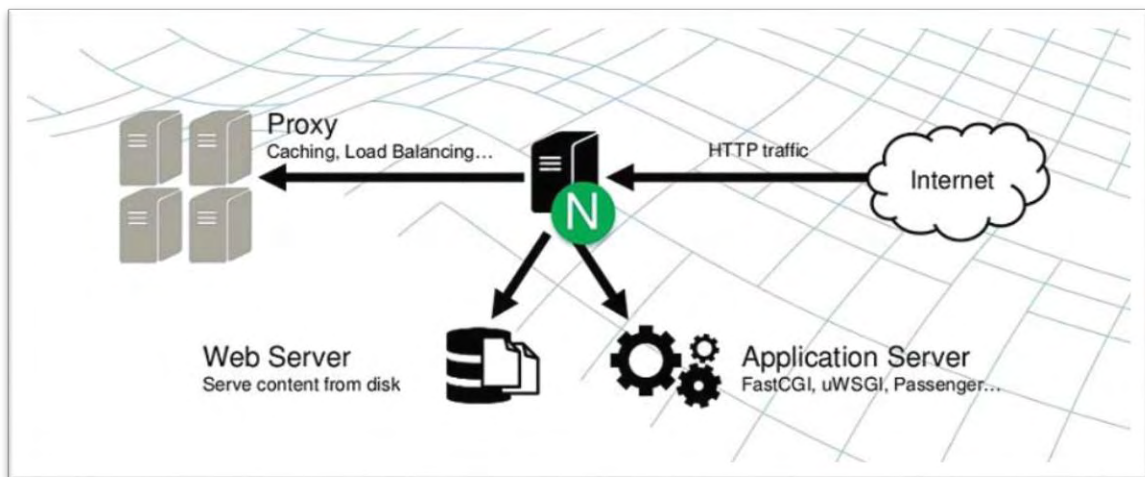


Εικόνα 3.4.1 Sencha Ext Js - Υποστήριξη όλων των συσκευών

Η λειτουργικότητα των πακέτων Sencha υποστηρίζεται σε όλα τα λειτουργικά και σε όλες τις συσκευές. Επιπλέον για τις συσκευές που υποστηρίζουν αφή, όπως τα κινητά τηλέφωνα και τα tablet υπάρχει και η λύση Sencha Touch [26]. Υποστηρίζει gestures και είναι βελτιστοποιημένο για μικρές οθόνες.

### 3.5. NGINX 1.11.11

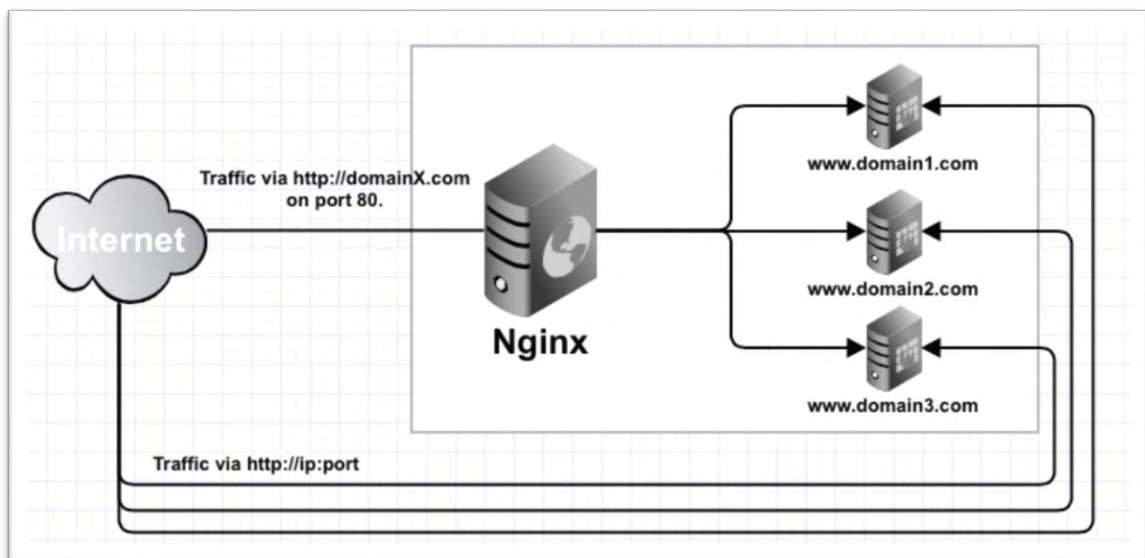
Το nginx [27] [28] (engine x) είναι ένας HTTP και reverse proxy server, ένας mail server καθώς επίσης και ένας proxy TCP/UDP generic server. Αρχικά το nginx αναπτύχθηκε από τον Igor Sysoev. Για πολλά χρόνια υποστήριζε την λειτουργία πολύ φορτωμένων ρωσικών ιστοχώρων όπως Yandex, Mail.Ru, VK και Rambler. Η συμμετοχή του στους web servers είναι πολύ μεγάλη και σύμφωνα με το Netcraft, για το 2017, αγγίζει το 29% στα πιο πολυσύχναστα sites. Ενδεικτικά μερικές επιτυχίες του είναι τα Netflix, Wordpress.com, FastMail.FM.



Εικόνα 3.5.1. Nginx - Βασική λειτουργία

Ενδεικτικά μερικά από τα βασικά χαρακτηριστικά του είναι τα παρακάτω:

- ❖ Εξυπηρετεί στατικά και index αρχεία, με autoindexing και open file descriptor cache
- ❖ Υποστηρίζει accelerated reverse proxying with caching, load balancing and fault tolerance;
- ❖ Υποστηρίζει accelerated support with caching of FastCGI, uwsgi, SCGI, and memcached servers; load balancing and fault tolerance;
- ❖ Modular αρχιτεκτονική. Τα φίλτρα περιλαμβάνουν gzipping, byte ranges, chunked responses, XSLT, SSI, και μετασχηματισμούς εικόνας. Πολλαπλά SSI inclusions σε μια σελίδα επεξεργάζονται παράλληλα, αν αυτά διαχειρίζονται από proxied η FastCGI/uwsgi/SCGI servers
- ❖ Υποστήριξη SSL και TLS SNI
- ❖ Υψηστήριξη για HTTP/2 με προτεραιότητες Βάρους και εξαρτήσεων.





### Εικόνα 3.5.1 Nginx - Παράδειγμα Reverse Proxy

Επιπλέον αξίζει να αναφερθούμε και σε κάποια πιο εξειδικευμένα χαρακτηριστικά όπως τα παρακάτω:

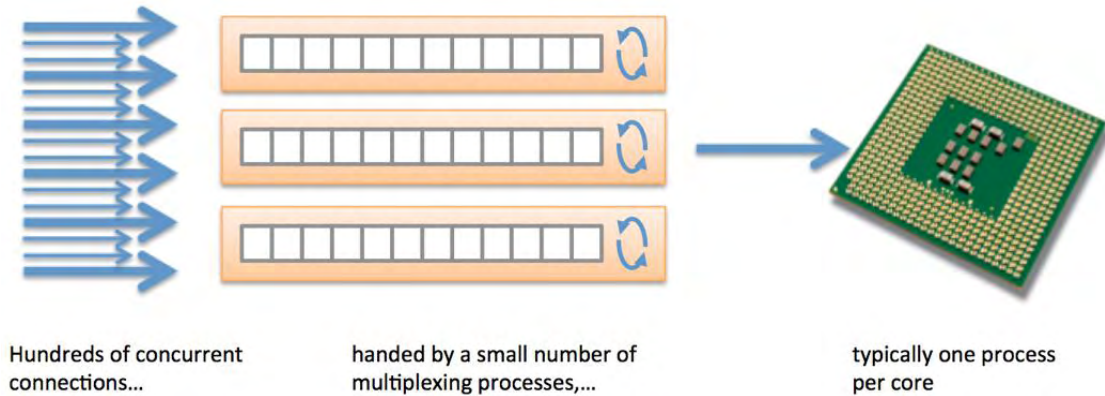
- ❖ Name-based και IP-based virtual servers;
- ❖ Keep-alive και pipelined υποστήριξη συνδέσεων
- ❖ Access log formats, buffered log writing, fast log rotation, and syslog logging
- ❖ 3xx-5xx error codes redirection
- ❖ The rewrite module: μεταβολή URI με χρήση regular expressions
- ❖ Εκτέλεση διαφορετικών διαδικασιών ανάλογα με την διεύθυνση του client
- ❖ Έλεγχος πρόσβασης βασισμένος σε client IP διευθύνσεις, με κωδικό (HTTP Basic authentication) και ανάλογα με το αποτέλεσμα της υπο-κλήσης
- ❖ Επικύρωση του HTTP referer
- ❖ Υποστήριξη μεθόδων PUT, DELETE, MKCOL, COPY, και MOVE
- ❖ FLV και MP4 streaming
- ❖ Response rate limiting
- ❖ Περιορισμός μέγιστου αριθμού συνδέσεων προερχόμενων από μια μοναδική διεύθυνση
- ❖ IP-based geolocation
- ❖ A/B testing
- ❖ Embedded Perl
- ❖ nginxScript

Α αρχιτεκτονικά του nginx ακολουθεί το μοντέλο της μιας κύριας διεργασίας και μερικών διεργασιών «εργατών» (worker processes), οι οποίες εκτελούνται στο περιβάλλον ενός χρήστη χωρίς δικαιώματα. Η φιλοσοφία που ακολουθείται στην παραμετροποίηση είναι η διάφανη και αδιάκοπη εξυπηρέτηση του client, σε περιπτώσεις αλλαγής παραμετροποίησης. Επιπλέον η μηχανή του υποστηρίζει τα

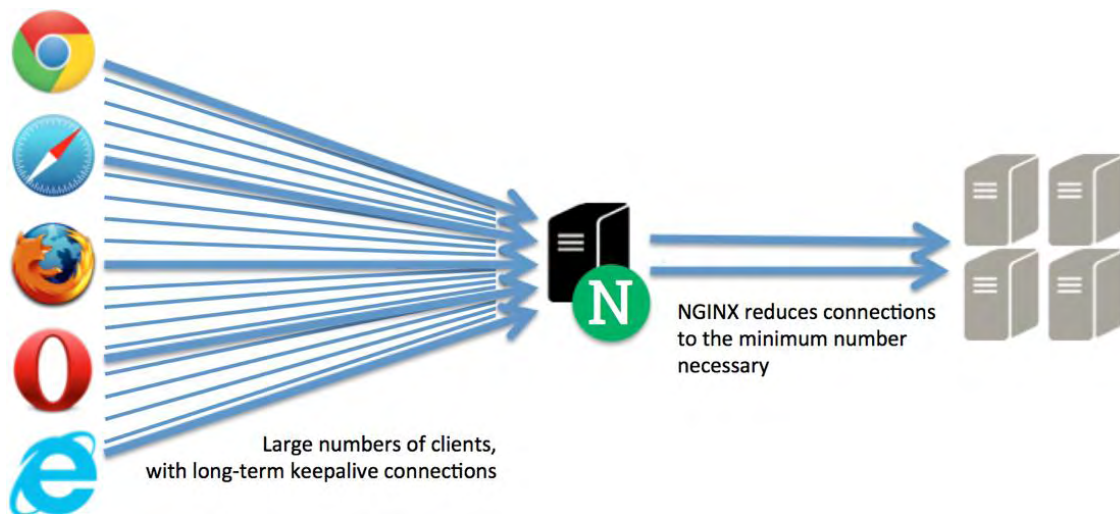
- ❖ kqueue (FreeBSD 4.1+), epoll (Linux 2.6+), /dev/poll (Solaris 7 11/99+), event ports (Solaris 10), select, και poll.
- ❖ Διάφορα kqueue χαρακτηριστικά περιλαμβανομένου τα EV\_CLEAR, EV\_DISABLE (προσωρινή απενεργοποίηση των event), NOTE\_LOWAT, EV\_EOF, πλήθος διαθέσιμων δεδομένων και κωδικών σφαλμάτων
- ❖ Διάφορα epoll χαρακτηριστικά περιλαμβανομένου τα EPOLLRDHUP (Linux 2.6.17+, glibc 2.8+) και EPOLLEXCLUSIVE (Linux 4.5+, glibc 2.24+)
- ❖ sendfile (FreeBSD 3.1+, Linux 2.2+, macOS 10.5+), sendfile64 (Linux 2.4.21+), και sendfilev (Solaris 8 7/01+)
- ❖ File AIO (FreeBSD 4.3+, Linux 2.6.22+)



- ❖ DIRECTIO (FreeBSD 4.4+, Linux 2.4+, Solaris 2.6+, macOS)
- ❖ Φίλτρα αποδοχής (FreeBSD 4.1+, NetBSD 5.0+) και TCP\_DEFER\_ACCEPT (Linux 2.4+)
- ❖ 10,000 ανενεργές συνδέσεις HTTP keep-alive καταλαμβάνουν περίπου 2.5M μνήμης
- ❖ Λειτουργίες αντιγραφής δεδομένων περιορίζονται στις απολύτως αναγκαίες.



Εικόνα 3.5.4. Nginx - Εξυπηρέτηση μεγάλου αριθμού συνδέσεων από λίγες διεργασίες



Εικόνα 3.5.5. Nginx - Εξυπηρέτηση μεγάλου αριθμού πελατών

### 3.6. Google Map API v3

Οι Χάρτες Google (Αγγλικά: Google Maps) είναι υπηρεσία χαρτογράφησης στο Διαδίκτυο. Η εφαρμογή και η τεχνολογία της υπηρεσίας παρέχεται από την Google και υποστηρίζει πολλές υπηρεσίες που βασίζονται σε χάρτες, συμπεριλαμβανομένου της ιστοσελίδας "Google Maps" [29]. Προσφέρει χάρτες δρόμων και σχεδιασθή διαδρομών για μεταφορές με τα πόδια, αυτοκίνητο, ποδήλατο ή μέσα μαζικής μεταφοράς. Περιλαμβάνει επίσης εντοπισμό των επιχειρήσεων που βρίσκονται σε πόλεις σε πολλές χώρες σε όλο τον κόσμο. Οι δορυφορικές εικόνες των Χαρτών Google δεν ανανεώνονται σε πραγματικό χρόνο, ωστόσο η Google

προσθέτει δεδομένα στη κύρια Βάση δεδομένων της σε τακτική βάση και οι περισσότερες από τις εικόνες δεν είναι πάνω από τριών ετών. Αξιοσημείωτο είναι ότι σε αναβάθμιση τον Ιούλιο του 2012, η Ελλάδα ήταν μία από τις 11 χώρες που επιλέχθηκαν.

Το Google maps API [30] υποστηρίζει τέσσερις διαφορετικού τύπους χαρτών τους:

**1. Δρόμοι (roadmap)**

Είναι η προεπιλεγμένη προβολή χαρτών και η μορφή της μοιάζει με τους οδικούς χάρτες.

**2. Δορυφόρος**

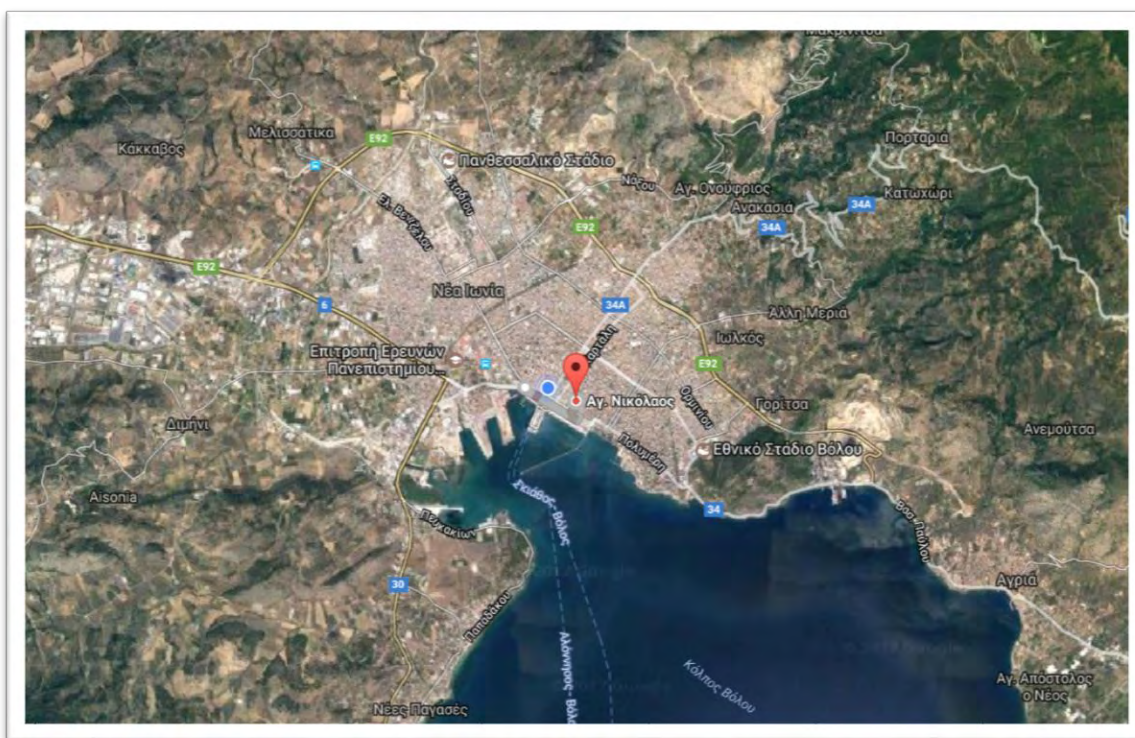
Παρουσιάζει φωτογραφία δορυφόρου στην προβαλλόμενη περιοχή

**3. Έδαφος**

Είναι μια μορφή ανάγλυφου που μας δίνει την αίσθηση ύψους των λόφων και βουνών της περιοχής.

**4. Υβριδικός**

Είναι ο χάρτης δορυφόρου μαζί με τον οδικό χάρτη.



Εικόνα 3.6.1 Google Maps - Υβριδικός

Η Google έχει αναπτύξει API για όλα τα λειτουργικά συστήματα και για όλες τις συσκευές. Παρακάτω γίνεται μια συνοπτική παρουσίασή τους :

❖ **Google Maps Android API**

Το πρώτο API αφορά στην ενσωμάτωση των Google maps σε Android συσκευές. Ο προγραμματισμός γίνεται με την χρήση Java. Η προτεινόμενη πλατφόρμα της Google για την ανάπτυξη εφαρμογών είναι το Android Studio [31]

#### ❖ Google Maps Javascript API

Το δεύτερο API είναι ίσως και το πιο ευρέως διαδεδομένο, μιας και είναι η υλοποίηση των Google maps με την χρήση Javascript. Με αυτόν τον τρόπο μπορεί ο προγραμματιστής να ενσωματώσει τις λειτουργίες Google maps σε οποιαδήποτε συσκευή με φυλλομετρητή.

#### ❖ Google Maps Geocoding API

Με το API αυτό έχει την δυνατότητα ο προγραμματιστής να εκτελέσει ερωτήματα Geocoding και Reverse Geocoding.

##### Geocoding

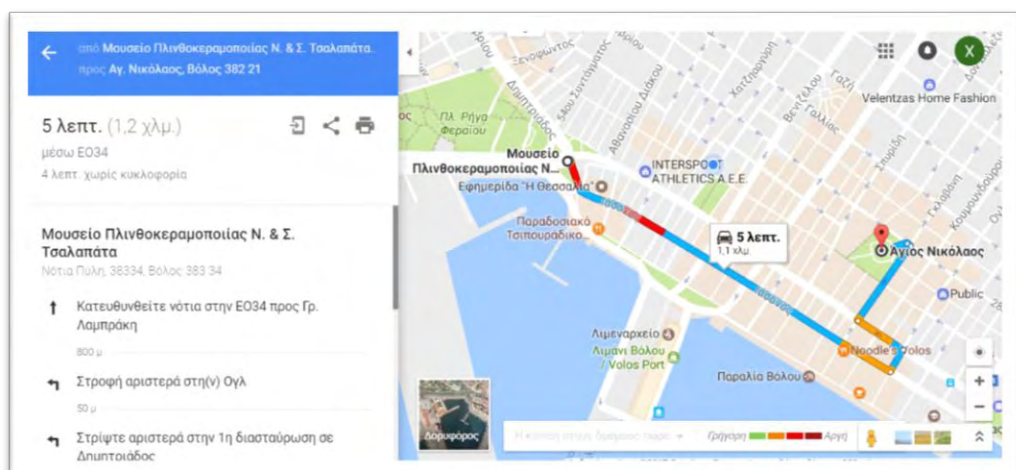
Είναι η διαδικασία κατά την οποία μια διεύθυνση μετατρέπεται σε γεωγραφικές συντεταγμένες (γεωγραφικό μήκος και πλάτος)

##### Reverse Geocoding

Είναι η διαδικασία κατά την οποία συντεταγμένες μετατρέπονται σε διευθύνσεις κατανοητές από ανθρώπους.

#### ❖ Google Maps Directions API

Με το Directions API μπορεί να παραχθεί μία διαδρομή, με αναλυτικές οδηγίες σε κείμενο κατάλληλο για ανάγνωση καθώς και οπτικό δρομολόγιο πάνω στον χάρτη. Έχει τις δυνατότητες ορισμού πολλαπλών σημείων προορισμού καθώς και δυνατότητα προσθήκης συνθηκών και σημείων αποφυγής (πχ διόδια). Η διαδρομή δημιουργείται ανάλογα με το μέσο (αυτοκίνητο, λεωφορείο, τρένο, πλοίο, ποδήλατο και άνθρωπος) και ο χρόνος υπολογίζεται βάση ιστορικών και τρεχόντων συνθηκών.



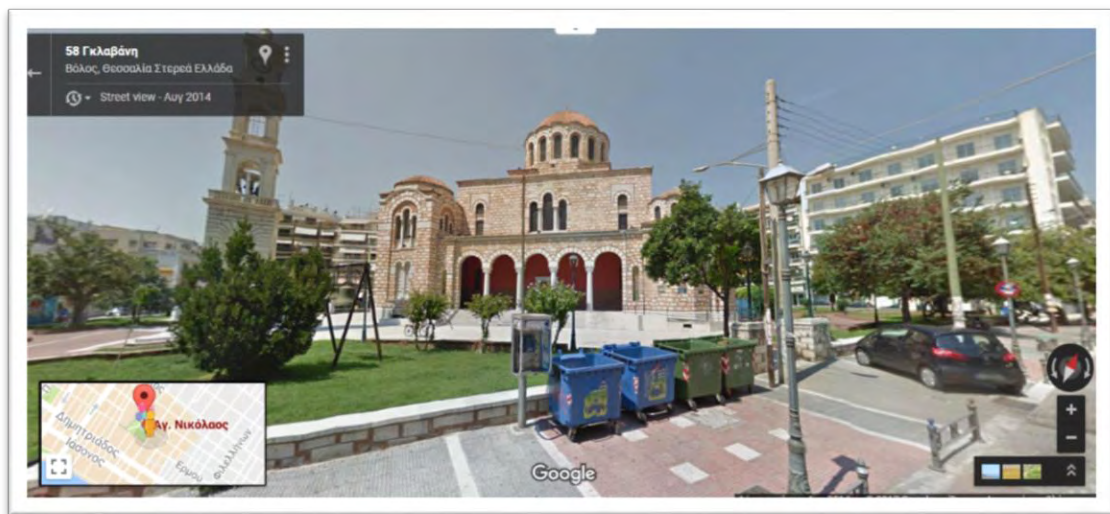
Εικόνα 3.6.2 Παράδειγμα Οδηγιών Google Maps

#### ❖ Google Places API Web Service



Το Places API συνδυάζει σημεία ενδιαφέροντος (POI points of interest) και επιχειρήσεις. Το πλήθος των σημείων ξεπερνά τα 100 εκατομμύρια και ενημερώνονται καθημερινά με την συνεισφορά των χρηστών.

Μια επιπλέον ενδιαφέρουσα λειτουργία των Google Maps είναι το Street View. Με αυτήν την επέκταση μπορεί κανείς να δει ένα σημείο του χάρτη σαν να βρισκόταν πραγματικά εκεί, πάνω στον δρόμο. Η λειτουργία αυτή είναι ενεργοποιημένη στους δρόμους από τους οποίους έχει διέλθει το χαρακτηριστικό αυτοκίνητο της Google με την 360 μοιρών κάμερά του.



Εικόνα 3.6.3 Street View

Η μεταπτυχιακή διατριβή βασίζεται στο Google Maps Javascript API για την προβολή των χαρτών, οι οποίοι εμπλουτίζονται με γεωγραφικά και γεωχρονικά ερωτήματα. Με την βοήθεια των Google Maps δημιουργούνται επιπλέον επίπεδα πάνω στον χάρτη και η εφαρμογή σχεδιάζει επάνω τους τα γεωγραφικά και γεωχρονικά ερωτήματα με την μορφή βασικών σχημάτων.

Για την λειτουργία του Google maps API πρέπει ο προγραμματιστής να αποκτήσει ένα κλειδί και να το ενσωματώσει σε κάποιο σημείο του κώδικά του. Το κλειδί μπορεί κανείς να το αποκτήσει από την σελίδα στην ανάλογη διεύθυνση [32], όπου συνδέεται με το google account του. Εκεί θα πρέπει να συνδέσει το κλειδί με τον ιστοχώρο του. Μια καλή πρακτική είναι στις διευθύνσεις HTTP να προσθέσει και το "localhost", ούτως ώστε να μπορέσει να αναπτύξει την εφαρμογή του ο developer τοπικά και όχι απομακρυσμένα σε άλλο υπολογιστή. Η χρέωση των Google maps εξαρτάται από την χρήση των εργαλείων, αλλά περιλαμβάνει και δωρεάν πλάνο για εφαρμογές με μικρές απαιτήσεις.

### 3.7. Insomnia Rest Client v5

Το Insomnia [33] είναι ίσως το καλύτερο δωρεάν πρόγραμμα cross platform για αποσφαλμάτωση restful υπηρεσιών. Περιλαμβάνει εύχρηστο περιβάλλον που βοηθά στην παραγωγικότητα του προγραμματιστή σε μεγάλο βαθμό. Η ανάπτυξη του Insomnia ξεκίνησε το 2015 από τον Gregory Schier. Η ενασχόλησή του με τα HTTP API τον ώθησε στην δημιουργία της εφαρμογής, διότι οι δοκιμές των API με το curl ήταν μη παραγωγικές και δύσκολες. Οι βασικές λειτουργίες του προγράμματος είναι οι παρακάτω :

#### 1. Δημιουργία HTTP κλήσης

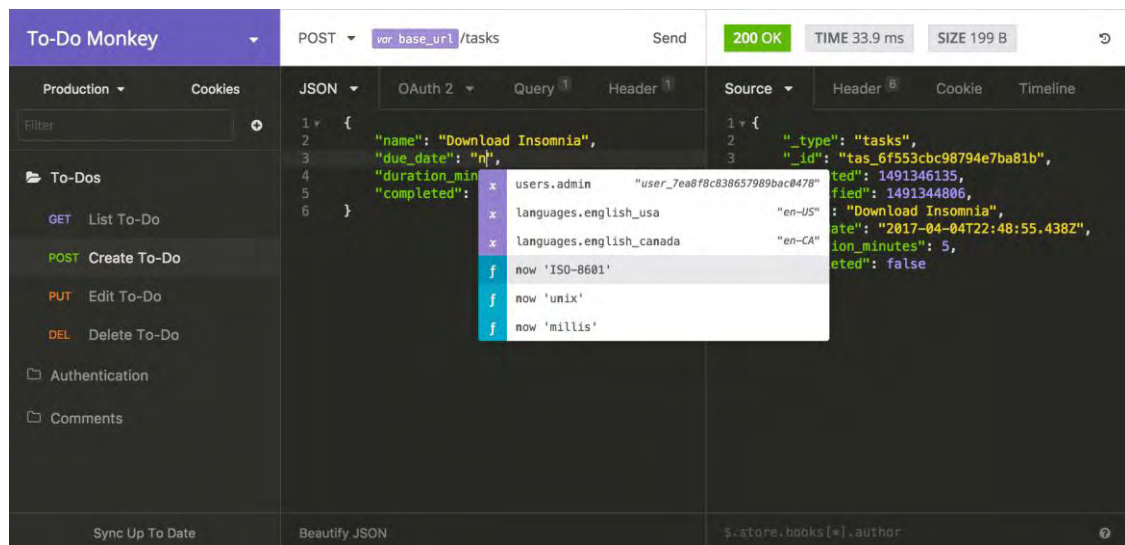
Μπορεί ο προγραμματιστής να ορίσει το URL τα δεδομένα, τις κεφαλίδες και το authorization σε ένα μόνο σημείο. Στην συνέχεια πατά το κουμπί “send” και εκτελείται η κλήση.

#### 2. Προβολή συναλλαγής αναλυτικά

Το πρόγραμμα μας πληροφορεί αναλυτικά για κάθε απάντηση. Μας δείχνει τα status code, body, headers, cookies και πολλά άλλα.

#### 3. Οργάνωση, Διαχείριση των κλήσεων

Δημιουργεί χώρους εργασίας (workspaces) και φακέλους (folders) με κλήσεις όπου μπορεί κανείς εύκολα με drag and drop να τα οργανώσει και επιπλέον να τα εξάγει ή εισάγει με αρχείο.



Εικόνα 3.7.1 Περιβάλλον Insomnia



*Βελέντζας Πολυχρόνης*  
*Διαχείριση Γεωγραφικών Ερωτημάτων και*  
*Υπηρεσία Διαδραστικών Χαρτών για Ιστοσελίδες*

## 4. Ανάπτυξη Πλατφόρμας

Η μεταπτυχιακή διατριβή αποτελείται από τις παρακάτω εφαρμογές:

1. Maps

Η “Maps” είναι ένας web server ο οποίος εξυπηρετεί την κυρίως ιστοσελίδα και τον σχεδιασμό των γεωγραφικών ερωτημάτων

2. MapsViewer

Η “MapsViewer” είναι ένας βοηθητικός web server που δημιουργήθηκε με σκοπό την υποστήριξη της υπηρεσίας προβολής των διαδραστικών χαρτών.

3. RestService

Αποτελεί την υπηρεσία REST API με την οποία μπορεί μια εξωτερική εφαρμογή να επικοινωνήσει με την βάση δεδομένων μας.

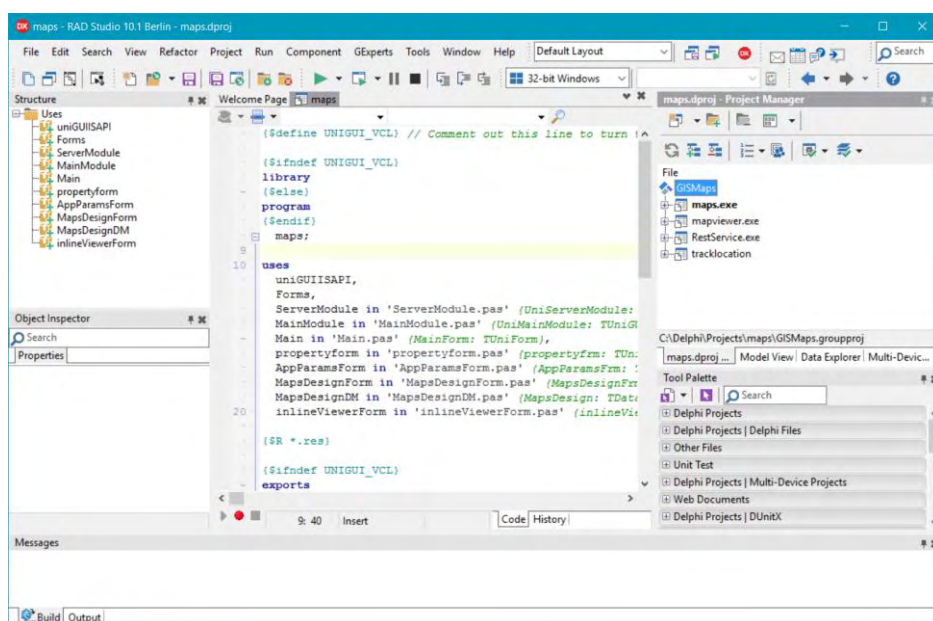
4. TrackLocation

Είναι μια βοηθητική εφαρμογή που συλλέγει στίγματα GPS.

5. Nginx Reverse Proxy

Χρησιμοποιείται ως reverse proxy για τα “Maps” και “RestService”

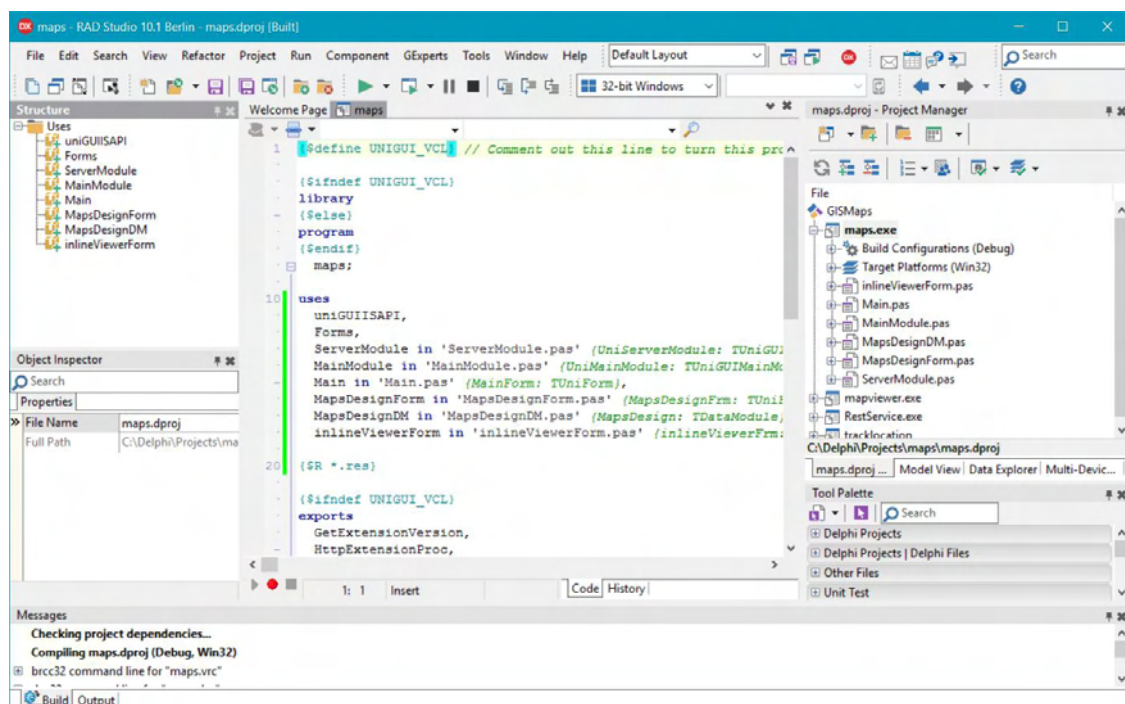
Το σύνολο των εφαρμογών που αναπτύχθηκαν χρησιμοποίησαν σαν βάση δεδομένων την PostgreSQL 9.5. Όλες οι εφαρμογές αναπτύχθηκαν με το Embarcadero Delphi XE 10.1 Berlin. Δημιουργήθηκε ένα Project Group με όνομα GISMaps και κατόπιν προστέθηκαν οι επιμέρους εφαρμογές.



Εικόνα 4.1 Περιβάλλον ανάπτυξης εφαρμογών Embarcadero Delphi XE 10.1 Berlin

## 4.1. Maps

Το Maps αποτελεί την βασική εφαρμογή της μεταπτυχιακής διατριβής. Παρέχει την ιστοσελίδα και την λειτουργικότητα της σχεδίασης των γεωγραφικών δεδομένων. Η ανάπτυξη της εφαρμογής έγινε χρησιμοποιώντας το Embarcadero Delphi με την επέκταση uniGUI. Για τους χάρτες επιλέχθηκε το Google maps API. Ανοίγοντας το Project Maps.exe εμφανίζονται οι φόρμες και modules δεδομένων της εφαρμογής, στο δεξιά πάνελ που είναι ο Project Manager.



Εικόνα 4.1.1 Project maps.exe

Το Project Maps.exe αποτελείται από τα παρακάτω forms και modules:

- ❖ ServerModule.pas

Είναι το module του server μας. Το unit αυτό περιέχει το αντικείμενο UniServerModule τύπου TUniServerModule. Εδώ έχουμε ορίσει τις εξής παραμέτρους του UniServerModule:

1. SessionTimeout

Είναι ο χρόνος σε χιλιοστά του δευτερολέπτου που διατηρείται ενεργό ένα session ενός φυλλομετρητή. Αν κατά την διάρκεια Nms δεν υπάρξει νέο request, τότε το session τερματίζεται. Επιλέξαμε τα 3600000ms, δηλαδή μία ώρα.

2. Title

Είναι ο τίτλος της εφαρμογής, εμφανίζεται ως τίτλος της ιστοσελίδας. Εισαγάγαμε τον τίτλο της μεταπτυχιακής διατριβής.

3. Charset



Είναι το προεπιλεγμένο charset του web server

#### 4. CustomCSS

Στο πεδίο αυτό τοποθετήσαμε το απαραίτητο CSS για την σωστή λειτουργία των χαρτών, εμφανίζεται παρακάτω:

```
<style type="text/css">
  html { height: 100% }
  body { height: 100%; margin: 0; padding: 0 }
  #uni_map_canvas { height: 100% }
</style>
```

#### 5. CustomFiles

Εδώ τοποθετούμε το Url που μας ενεργοποιεί το Google API :

```
//maps.googleapis.com/maps/api/js?key=AlzaSyCRbWYXCdxWu5pkXikDL-
N8Wt7gnktoXhU&sensor=false
```

Δεν ορίζουμε το πρωτόκολλο (HTTP ή HTTPS) για να το συμπληρώσει αυτόματα ο φυλλομετρητής σε περίπτωση ασφαλούς σύνδεσης. Στο Url προσθέτουμε και το αντίστοιχο κλειδί που μας παρέχει η Google.

#### 6. CustomMeta

Προσθέτουμε το παρακάτω:

```
<meta name="viewport" content="initial-scale=1.0, user-scalable=no" />
```

Είναι απαραίτητο για την σωστή λειτουργία του Google Maps

#### 7. MainFormDisplay

Καθορίζει τον τρόπο που θα εμφανισθεί η βασική μας φόρμα στον φυλλομετρητή.

Εδώ ορίσαμε το mfPage, που σημαίνει ότι η βασική μας φόρμα θα επεκτείνεται σε όλη την σελίδα.

#### ❖ MainModule.pas

Αυτό το module δημιουργείται για κάθε ένα session του web server. Η λειτουργία του είναι προσανατολισμένη στην επικοινωνία με την βάση δεδομένων. Κατά την δημιουργία του καλείται το OnCreate Event: UniGUIMainModuleCreate το οποίο εκτελεί ορισμένες απαραίτητες λειτουργίες για την σύνδεση στην βάση :

```
procedure TUniMainModule.UniGUIMainModuleCreate(Sender: TObject);
var fn,dir,app:string;
begin
  //App:Το αρχείο με την πλήρη διεύθυνση του εκτελέσιμου web server
  app:=ParamStr(0);
  //dir:Το path του εκτελέσιμου web server
  dir:=ExtractFileDir(app);
  fn:=ChangeFileExt(app,'.dat');
  //Ορίζουμε το filename του πίνακα ParamsTable
  ParamsTable.FileName:=fn;
  //Έλεγχος ύπαρξης αρχείου παραμέτρων
  if FileExists(fn) then begin
    ParamsTable.LoadFromFile(fn,sfJSON);
    //Διαβάζουμε τις παραμέτρους σύνδεσης της βάσης
    FDConnection.Params.Text:=ParamsTableconnstring.AsString;
    try
      //Συνδεόμαστε με την βάση
```





```
FDConnection.Connected:=true;  
except  
  on e:exception do  
    //Σε περίπτωση λάθους εμφανίζουμε το μήνυμα λάθους  
    showmessage(e.Message);  
  end;  
end  
else  
  ParamsTable.CreateDataSet;  
end;
```

Περιέχει τα εξής data components :

1. FDConnection

To FDConnection είναι η σύνδεση με την βάση δεδομένων.

2. ParamsTable

Πίνακας παραμέτρων με πληροφορίες σύνδεσης στην βάση δεδομένων (φορτώνεται από αρχείο)

3. LoginUser

Query με το ακόλουθο statement:

```
select *  
from wuser u  
where u.username=:username  
and u.password=:password
```

Τα ονόματα μεταβλητών που αρχίζουν από “:” είναι bind variables. Τα συμπληρώνουμε πριν την εκτέλεση του ερωτήματος. Ακολουθούμε αυτή την μεθοδολογία στα queries (με την εισαγωγή bind variables) διότι βοηθάμε την βάση να καταλάβει ότι πρόκειται για το ίδιο ερώτημα και ελαχιστοποιείται ο χρόνος parsing και ο χρόνος επιλογής πλάνου εκτέλεσης.

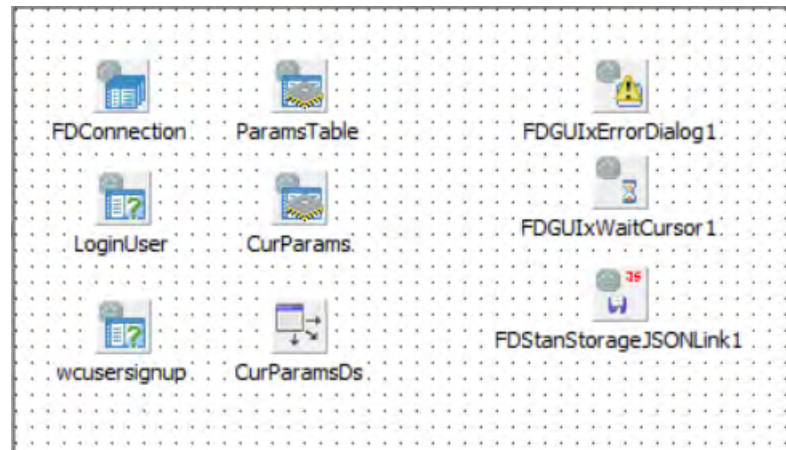
Το παραπάνω query μας επιστρέφει τα στοιχεία του χρήστη, σε περίπτωση επιτυχημένου login.

4. CurParams

Πίνακας στην μνήμη όπου αποθηκεύουμε πληροφορίες για τον χρήστη.

5. Wusersignup

6. Query που χρησιμοποιείται για να εισάγουμε στον πίνακα wuser μια νέα εγγραφή, ουσιαστικά ένα νέο account.



Εικόνα 4.1.2 Τα συστατικά του UniMainModule

Στο UniMainModule έχουν υλοποιηθεί ορισμένες βασικές procedure και function που χρησιμοποιούνται εκτεταμένα στην εφαρμογή όπως οι παρακάτω :

#### 1. hashString

Η hashString δέχεται στην είσοδό της ένα αλφαριθμητικό και μας επιστρέφει το hashed ισοδύναμό του. Το hashing πραγματοποιείται με τον Digest 5 αλγόριθμο. Την χρησιμοποιούμε για να μετατρέψουμε το κλειδί του χρήστη και με το hashed κλειδί να πραγματοποιούμε το απαραίτητο authentication.

```
function TUniMainModule.hashString(astr:widestring):widestring;  
var hashMessageDigest5 : TIdHashMessageDigest5;  
begin  
    hashMessageDigest5:= nil;  
    try  
        hashMessageDigest5:=TIdHashMessageDigest5.Create;  
        Result:=IdGlobal.IndyLowerCase ( hashMessageDigest5.HashStringAsHex (   
astr ) );  
    finally  
        hashMessageDigest5.Free;  
    end;  
end;
```

#### 2. getQuery

Με την getQuery μπορούμε να δημιουργήσουμε ένα dataset, δίνοντας ως είσοδο ένα sql statement και τις παραμέτρους του, οι οποίες περνούν στην βάση ως bind variables.

```
function TUniMainModule.getQuery(const SQL: WideString; Params: Variant):  
TDataset;  
var i,j:integer;  
    q:TFDQuery;  
begin  
    q:=TFDQuery.Create(self);  
    q.Connection:=FDConnection;  
    q.UpdateOptions.ReadOnly:=true;  
    q.sql.Text:=sql;  
    if q.ParamCount>0 then begin  
        if not varisarray(Params) then begin  
            if (not varisnull(Params)) or (not varisempty(Params)) then begin  
                q.Params[0].DataType:=getParamTypeFromVariant(params);  
                q.Params[0].Value:=Params;  
            end;  
        end;  
    end;
```





```
end;  
end  
else begin  
    for i:=0 to q.ParamCount-1 do begin  
        q.Params[i].DataType:=getParamTypeFromVariant(params[i]);  
        q.Params[i].Value:=Params[i];  
    end;  
end;  
end;  
try  
    q.Active:=true;  
    result:=q;  
except  
    On e:Exception do begin  
        raise Exception.Create('Query :'+sql+#13#10+' Exception Message  
: '+e.Message);  
    end;  
end;  
end;  
end;
```

### 3. execQuery

Έχει παρόμοια λειτουργία με την getQuery, με την διαφορά ότι εκτελεί ένα DML ή DDL statement στην βάση. Στην περίπτωση ενός insert, delete ή update statement επιστρέφει το πλήθος των γραμμών που επηρεάστηκαν.

### 4. sendmail\_ssl

Η παραπάνω procedure στέλνει email. Στην είσοδό της εισάγουμε τον αποστολέα, παραλήπτες, θέμα και το κυρίως κείμενο. Έχει υλοποιηθεί με το Ole αντικείμενο “CDO.Message”:

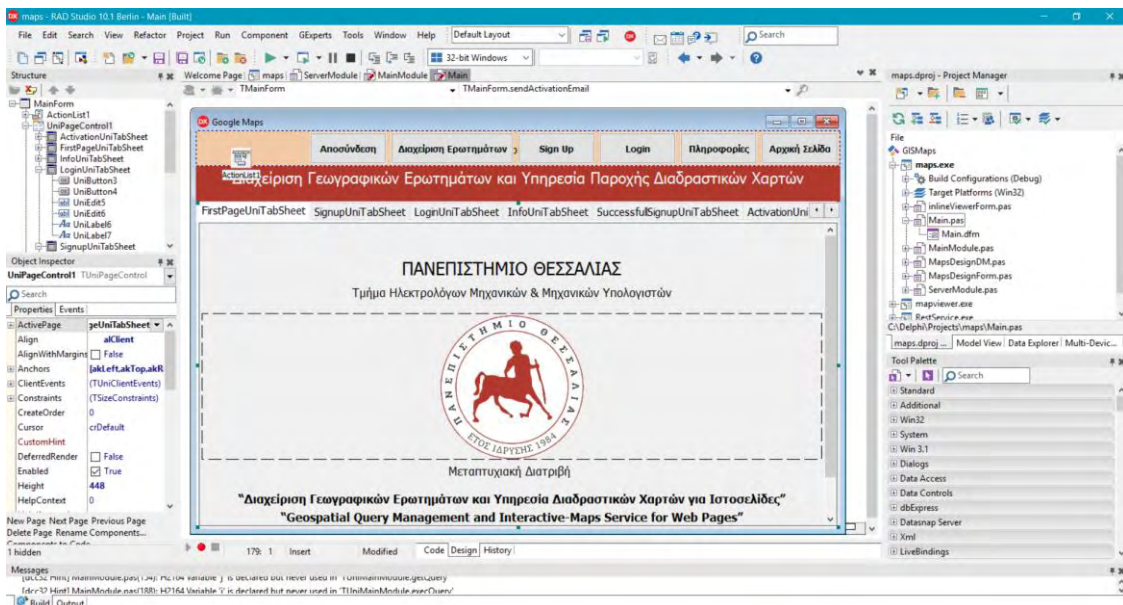
```
procedure TUniMainModule.sendmail_ssl(from, recipients, subject, body:string);  
var objMessage:OleVariant;  
begin  
    objMessage := CreateOleObject('CDO.Message');  
    objMessage.Subject:=subject;  
    objMessage.From:=from;  
    objMessage.To:=recipients;  
    objMessage.HTMLBody:=body;  
    objMessage.HTMLBodyPart.Charset:= 'windows-1253';  
  
    objMessage.Configuration.Fields.Item('http://schemas.microsoft.com/cdo/configuration/sendusing'):= 2 ;  
  
    objMessage.Configuration.Fields.Item('http://schemas.microsoft.com/cdo/configuration/smtpserver') := 'smtp.uth.gr';  
  
    objMessage.Configuration.Fields.Item('http://schemas.microsoft.com/cdo/configuration/smtpauthenticate') := 1;  
  
    objMessage.Configuration.Fields.Item('http://schemas.microsoft.com/cdo/configuration/sendusername') := 'cvelentzas';  
  
    objMessage.Configuration.Fields.Item('http://schemas.microsoft.com/cdo/configuration/sendpassword') := 'apassword';  
  
    objMessage.Configuration.Fields.Item('http://schemas.microsoft.com/cdo/configuration/smtpserverport') := 465 ;
```

```
objMessage.Configuration.Fields.Item('http://schemas.microsoft.com/cdo/configuration/smtpusessl') := True ;

objMessage.Configuration.Fields.Item('http://schemas.microsoft.com/cdo/configuration/smtpconnectiontimeout') := 60 ;
objMessage.Configuration.Fields.Update;
objMessage.Send;
end;
```

#### ❖ Main.pas

Το Unit περιέχει την φόρμα MainForm με την οποία έχουμε υλοποιήσει την ιστοσελίδα μας. Επιπλέον έχουμε υλοποιήσει όλες τις απαραίτητες διαδικασίες για signup (εγγραφή), login (σύνδεση) και logout (αποσύνδεση).



Εικόνα 4.1.3 Σχεδίαση Φόρμας MainForm

Η MainForm περιέχει μεταξύ άλλων τις εξής διαδικασίες:

#### 1. Διαδικασία signup

```
procedure TMainForm.UniButton1Click(Sender: TObject);
var p:integer;
begin
  if UniEdit3.Text='' then begin
    MessageDlg('Προσοχή δεν συμπληρώσατε το "Email"!',mtWarning,[mbOk] );
    exit;
  end;
  if UniEdit4.Text='' then begin
    MessageDlg('Προσοχή δεν συμπληρώσατε το "Ονοματεπώνυμο"!',mtWarning,[mbOk] );
    exit;
  end;
  if UniEdit1.Text='' then begin
    MessageDlg('Προσοχή δεν συμπληρώσατε το "Κωδικός"!',mtWarning,[mbOk] );
    exit;
```



**Βελέντζας Πολυχρόνης**  
**Διαχείριση Γεωγραφικών Ερωτημάτων και**  
**Υπηρεσία Διαδραστικών Χαρτών για Ιστοσελίδες**

```
end;
if UniEdit2.Text='' then begin
    MessageDlg('Προσοχή δεν συμπληρώσατε το "Επιβεβαίωση Κωδικού"',mtWarning,[mbOk] );
    exit;
end;
if UniEdit1.Text<>UniEdit2.Text then begin
    MessageDlg('Προσοχή δεν ταυριάζουν οι κωδικοί που εισάγατε!',mtWarning,[mbOk] );
    exit;
end;
if UniMainModule.getQueryVariant('select count(1) from wcuser where upper(email)=upper(:1) and isactive=1',UniEdit3.Text)>0 then begin
    MessageDlg('Προσοχή, υπάρχει χρήστης με το ίδιο email!',mtWarning,[mbOk] );
    exit;
end;
//Έλεγχος ορθότητας email
p:=Pos('.',UniEdit3.Text);
if p=0 then begin
    MessageDlg('Προσοχή, λάθος email!',mtWarning,[mbOk] );
    exit;
end;

p:=Pos('@',UniEdit3.Text);
if p=0 then begin
    MessageDlg('Προσοχή, λάθος email!',mtWarning,[mbOk] );
    exit;
end;

//Σύνδεση στην βάση
UniMainModule.FDConnection.Connected:=true;

//Διαγραφή όμοιων email που δεν έχουν ενεργοποιηθεί
UniMainModule.execQuery('delete from wcuser where upper(email)=upper(:1) and isactive=0',UniEdit3.Text);

UniMainModule.wcusersignup.Active:=true;
UniMainModule.wcusersignup.Edit;

//Εισαγωγή στοιχείων χρήστη
UniMainModule.wcusersignupemail.AsWideString:=UniEdit3.Text;
UniMainModule.wcusersignupusername.AsWideString:=copy(UniEdit3.Text,1,p-1);
UniMainModule.wcusersignuprealname.AsWideString:=UniEdit4.Text;
UniMainModule.wcusersignupisactive.AsInteger:=0;

UniMainModule.wcusersignuppassword.AsWideString:=UniMainModule.hashString(LowerCase(uniedit1.Text));
if UniCheckBox1.Checked then
    UniMainModule.wcusersignupdodemo.AsInteger:=1
else
    UniMainModule.wcusersignupdodemo.AsInteger:=0;

//Εισαγωγή στην βάση, η Post στέλνει τις αλλαγές στην βάση και κάνει commit
UniMainModule.wcusersignup.Post;

//Αποσύνδεση
UniMainModule.FDConnection.Connected:=false;

//Αποστολή email ενεργοποίησης
sendActivationEmail(UniEdit3.Text);
UniLabel12.Text:=UniEdit3.Text;
UniPageControl1.ActivePageIndex:=5;
```

end;

## 2. Διαδικασία Ενεργοποίησης Λογαριασμού

```
procedure TMainForm.UniButton5Click(Sender: TObject);
var id:integer;
begin
  if UniEdit7.Text='' then begin
    MessageDlg('Εισάγετε κωδικό ενεργοποίησης!',mtWarning,[mbOk] );
    exit;
  end;
  id:=UniMainModule.getQueryInteger('select id from wcuser where
upper(email)=upper(:1)',UniEdit3.Text);

  if id=0 then begin
    showMessage('Προσοχή το email δεν βρέθηκε!');
    exit;
  end;

  //Εύρεση μη ενεργοποιημένης εγγραφής
  id:=UniMainModule.getQueryInteger('select id from wcuser where
upper(email)=upper(:1)
activationcode=:2',vararrayof([UniEdit3.Text,uniedit7.Text]));
  if id=0 then begin
    showMessage('Λάθος κωδικός ενεργοποίησης!');
    exit;
  end;

  //Σύνδεση στην βάση
  UniMainModule.FDConnection.Connected:=true;

  //Ξεκινά ένα transaction, σε περίπτωση προβλήματος θα γίνει rollback
  UniMainModule.FDConnection.StartTransaction;
  try
    //Ενεργοποίηση λογαριασμού και είσοδος στην βάση
    UniMainModule.execQuery('update wcuser set isactive=1 where id=:1',id);
    UniMainModule.execQuery('insert into wcloggedin
(usrid,sessionid,endpoint)
values(:1,:2,'web')',vararrayof([id,UniSession.SessionId]));
    checklogin();
    //Δημιουργία Βάσης Χρήστη
    createDatabase(UniMainModule.CurParamsemail.AsWideString,
                  UniMainModule.CurParamshashedpassword.AsWideString,
                  UniMainModule.getQueryInteger('select dodemo from wcuser
where id=:1',id)
                );
    UniMainModule.FDConnection.Commit;
  except
    on e:exception do begin
      ShowMessage('Προσοχή σφάλμα 101!');
      //Rollback, Οι αλλαγές ακυρώνονται
      UniMainModule.FDConnection.Rollback;
      exit;
    end;
  end;
  //Αποσύνδεση
  UniMainModule.FDConnection.Connected:=false;
  UniPageControl1.ActivePage:=SuccessfulSignUpUniTabSheet;
  //Έλεγχος login
  checklogin();
end;
```

## 3. Διαδικασία login



```
procedure TMainForm.UniButton3Click(Sender: TObject);
var id:Integer;
begin
  //Εύρεση id χρήστη
  id:=UniMainModule.getQueryInteger('select id from wcuser where
upper(email)=upper(:1) and
upper(password)=upper(:2)',vararrayof([UniEdit5.Text,UniMainModule.hashString(uniedit6.Text)]));
  if id=0 then begin
    ShowMessage('Λάθος στοιχεία σύνδεσης!');
    exit;
  end;
  //Σύνδεση στην βάση
  UniMainModule.FDConnection.Connected:=true;
  UniMainModule.executeQuery('delete from wcloggedin where usrid=:1 and
endpoint='web'',id);
  UniMainModule.executeQuery('insert into wcloggedin (usrid,sessionid,endpoint)
values(:1,:2,'web')',vararrayof([id,UniSession.SessionId]));
  checklogin();
  //Αποσύνδεση
  UniMainModule.FDConnection.Connected:=false;
  UniPageControl1.ActivePage:=SuccessfullLoginUniTabsheet;
end;
```

#### 4. Διαδικασία logout

```
procedure TMainForm.UniSpeedButton6Click(Sender: TObject);
begin
  //Διαγραφή από τον πίνακα wcloggedin
  UniMainModule.executeQuery('delete from wcloggedin where usrid=:1 and
endpoint='web'',UniMainModule.usrid);
  //Ανανέωση της σελίδας
  UniApplication.Restart;
end;
```

#### 5. Δημιουργία Βάσης

```
procedure
TMainForm.createDatabase(aemail,apassword:widestring;adodemo:integer);
begin
  //Δημιουργία Ρόλου Χρήστη
  UniMainModule.executeQuery('CREATE ROLE "user_'+aemail+'" LOGIN PASSWORD
''+apassword+'' NOINHERIT VALID UNTIL 'infinity''');
  //Δημιουργία Schema Χρήστη
  UniMainModule.executeQuery('CREATE SCHEMA "user_'+aemail+'" AUTHORIZATION
"user_'+aemail+'''');
  //Δικαιώματα Ρόλου χρήστη στο Schema
  UniMainModule.executeQuery('GRANT ALL ON SCHEMA "user_'+aemail+'" TO
"user_'+aemail+'''');
  UniMainModule.executeQuery('GRANT USAGE ON SCHEMA Public TO
"user_'+aemail+'''');
  //Δημιουργία απαραίτητων πινάκων wcqueries, wcparams
  UniMainModule.executeQuery('CREATE TABLE "user_'+aemail+'".wcqueries('+
'id integer NOT NULL,'+
'usrid integer,'+
'caption character varying(50),'+
'color character varying(10),'+
'sql text,'+
'geomcolumn character varying(50),'+
'isvisible smallint DEFAULT 1,'+
'linenum integer,'+
'descrcolumn character varying(50),'+
'CONSTRAINT wcqueries_pk PRIMARY KEY (id)'+
')');
```



```
        'WITH ('+
        ' OIDS=FALSE'+
        '),'');
UniMainModule.executeQuery('CREATE                                     SEQUENCE
"user_'+aemail+'".wcqueries_seq','');
UniMainModule.executeQuery('ALTER TABLE "user_'+aemail+'".wcqueries OWNER TO
"user_'+aemail+'"'','');
UniMainModule.executeQuery('ALTER sequence "user_'+aemail+'".wcqueries_seq
OWNER TO "user_'+aemail+'"'','');

UniMainModule.executeQuery('CREATE TABLE "user_'+aemail+'".wparams '+
'(' '+
' id serial, '+
' paramname character varying(50), '+
' datatype smallint, '+
' wherepart character varying(255), '+
' caption character varying(50), '+
' valuesquery character varying(255), '+
' defaultvalue character varying(255), '+
' minvalue character varying(255), '+
' maxvalue character varying(255), '+
' linenum integer, '+
' PRIMARY KEY (id) '+
') '+
'WITH ( '+
' OIDS = FALSE '+
')','');
UniMainModule.executeQuery('CREATE                                     SEQUENCE
"user_'+aemail+'".wparams_seq','');

//Αλλαγή owner των sequence
UniMainModule.executeQuery('ALTER TABLE "user_'+aemail+'".wparams OWNER TO
"user_'+aemail+'"'','');
UniMainModule.executeQuery('ALTER sequence "user_'+aemail+'".wparams_id_seq
OWNER TO "user_'+aemail+'"'','');

//Αν ο χρήστης έχει ζητήσει demo δεδομένα, τα δημιουργούμε από τον demo
χρήστη
if adodemo=1 then begin
    UniMainModule.executeQuery('insert                                     into
"user_'+aemail+'".wcqueries(id,usrid,caption,color,sql,geomcolumn,isvisible,
linenum,descrcolumn) '+
                                'select
nextval(''wcqueries_seq''),:1,caption,color,sql,geomcolumn,isvisible,linenum
,descrcolumn '+
                                'from demo.wcqueries',
                                UniMainModule.CurParamsusrid.AsInteger);
    UniMainModule.executeQuery('CREATE TABLE "user_'+aemail+'".coasts (LIKE
demo.coasts INCLUDING CONSTRAINTS INCLUDING INDEXES)','');
    UniMainModule.executeQuery('CREATE TABLE "user_'+aemail+'".municipalities
(LIKE demo.municipalities INCLUDING CONSTRAINTS INCLUDING INDEXES)','');
    UniMainModule.executeQuery('CREATE TABLE "user_'+aemail+'".natura (LIKE
demo.natura INCLUDING CONSTRAINTS INCLUDING INDEXES)','');
    UniMainModule.executeQuery('CREATE TABLE "user_'+aemail+'".peripheries (LIKE
demo.peripheries INCLUDING CONSTRAINTS INCLUDING INDEXES)','');
    UniMainModule.executeQuery('CREATE TABLE "user_'+aemail+'".places (LIKE
demo.places INCLUDING CONSTRAINTS INCLUDING INDEXES)','');
    UniMainModule.executeQuery('CREATE TABLE "user_'+aemail+'".waterways (LIKE
demo.waterways INCLUDING CONSTRAINTS INCLUDING INDEXES)','');
    UniMainModule.executeQuery('insert into "user_'+aemail+'".coasts select *
from demo.coasts','');
    UniMainModule.executeQuery('insert into "user_'+aemail+'".municipalities
select * from demo.municipalities','');
```

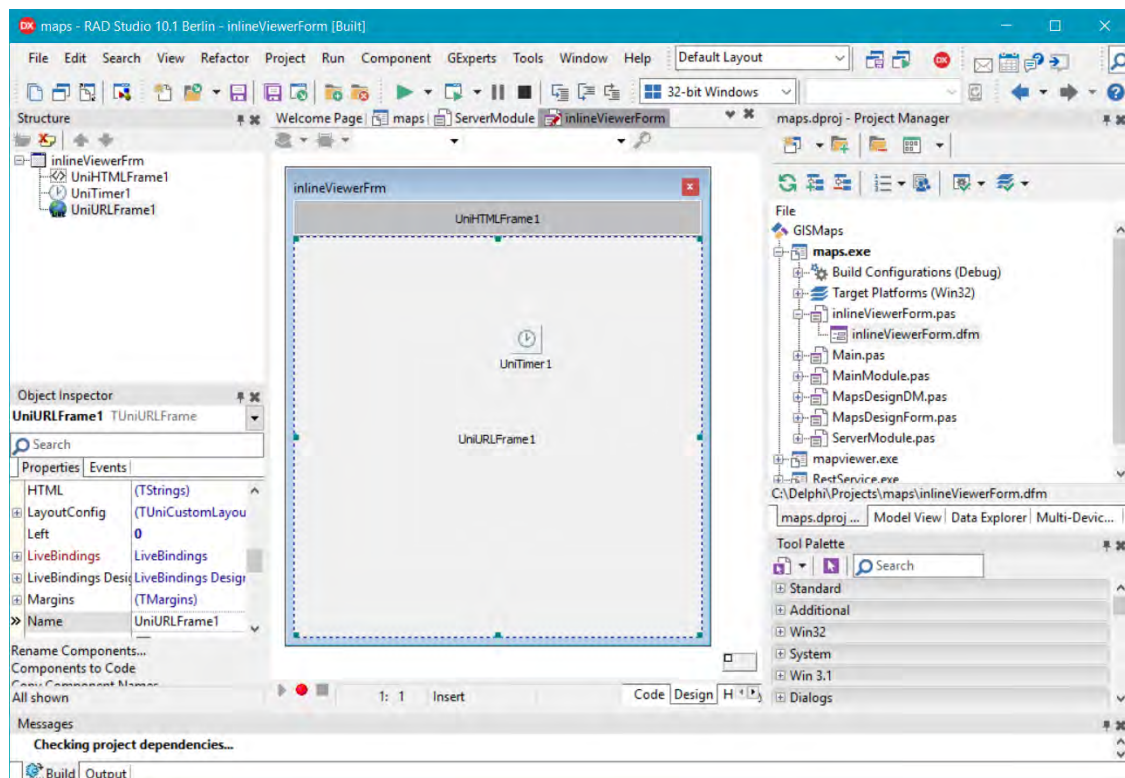


```
UniMainModule.executeQuery('insert into "user_'+aemail+'".natura select *
from demo.natura','');
UniMainModule.executeQuery('insert into "user_'+aemail+'".peripheries select
* from demo.peripheries','');
UniMainModule.executeQuery('insert into "user_'+aemail+'".places select *
from demo.places','');
UniMainModule.executeQuery('insert into "user_'+aemail+'".waterways select *
from demo.waterways','');

UniMainModule.executeQuery('ALTER TABLE "user_'+aemail+'".coasts OWNER TO
"user_'+aemail+'','');
UniMainModule.executeQuery('ALTER TABLE "user_'+aemail+'".municipalities
OWNER TO "user_'+aemail+'','');
UniMainModule.executeQuery('ALTER TABLE "user_'+aemail+'".natura OWNER TO
"user_'+aemail+'','');
UniMainModule.executeQuery('ALTER TABLE "user_'+aemail+'".peripheries OWNER
TO "user_'+aemail+'','');
UniMainModule.executeQuery('ALTER TABLE "user_'+aemail+'".places OWNER TO
"user_'+aemail+'','');
UniMainModule.executeQuery('ALTER TABLE "user_'+aemail+'".waterways OWNER TO
"user_'+aemail+'','');
end;
end;
```

❖ inlineViewerForm.pas

Το inlineViewerForm είναι ένα unit που περιέχει την φόρμα προεπισκόπησης κινητού. Δεν έχει κάποια λειτουργικότητα παρά μόνο την προβολή του χάρτη μέσω του component UniURLFrame. Η σχεδίαση της φόρμας του εμφανίζεται παρακάτω:



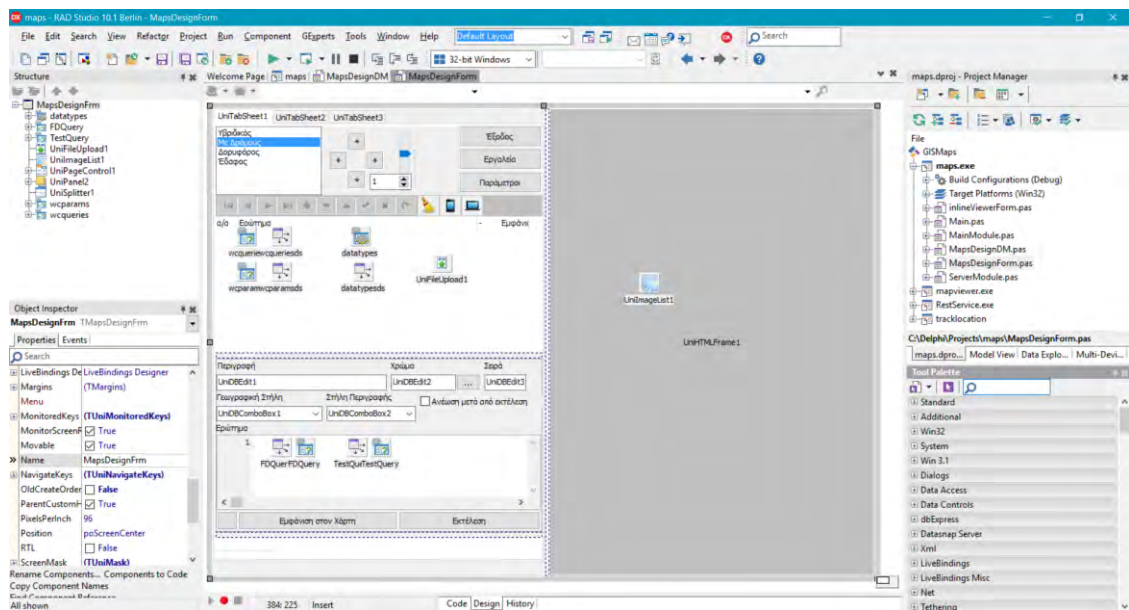
Εικόνα 4.1.4 Φόρμα inlineViewerFrm

#### ❖ MapsDesignDM.pas

Το Unit αυτό περιλαμβάνει το datamodule “MapsDesign”. Είναι υπεύθυνο για την σύνδεση με την βάση δεδομένων του χρήστη. Έχει παρόμοια λειτουργικότητα με το UniMainModule και σε αυτό έχουμε επίσης υλοποιήσει τα “getQuery” και “execQuery”.

#### ❖ MapsDesignForm.pas

Το αρχείο αυτό περιέχει την φόρμα “MapsDesignFrm” όπου λαμβάνει χώρα η διαχείριση των γεωγραφικών μας ερωτημάτων. Κάθε φορά που δημιουργείται ένα αντικείμενο “MapsDesignFrm” δημιουργούμε και το αντίστοιχο datamodule “MapsDesign” για να εξυπηρετήσουμε τις κλήσεις προς την βάση δεδομένων.



Εικόνα 4.1.5 Φόρμα MapsDesign

Σε αυτό μπορούμε να δούμε το σύνολο των controls με τα οποία έχει κατασκευασθεί στην προηγούμενη εικόνα. Μεταξύ των controls υπάρχει το “UniHTMLFrame1” στο οποίο έχουμε ενσωματώσει τον χάρτη Google. Για να το καταφέρουμε αυτό εισάγουμε το ακόλουθο κομμάτι html στο property “HTML” του “UniHTMLFrame1” :

```
<div id="uni_map_canvas" style="width: 100%; height: 100%"></div>
```

Με τον τρόπο αυτό έχουμε δημιουργήσει το απαιτούμενο “div” html tag στο οποίο στην συνέχεια θα χρησιμοποιήσει το Google Maps API για το rendering του χάρτη. Για να υποδείξουμε ότι αυτό είναι το σημείο με τον χάρτη θα χρησιμοποιήσουμε το παρακάτω javascript:

```
var umap = document.getElementById("uni_map_canvas");  
var map = new google.maps.Map(umap, myOptions);
```

Μετά το άνοιγμα της φόρμας “UniHTMLFrame1” ενεργοποιείται το Event UniHTMLFrame1.ClientEvents.afterupdatehtml. Αυτό το Event αφαιρεί τον χάρτη μας και προσθέτει την απαραίτητη λειτουργικότητα στον χάρτη. Συγκεκριμένα περιέχει

διαδικασίες για την δημιουργία πρόσθετων επιπέδων (Layers), για τον σχεδιασμό πινέζας (Pin), για την διαχείριση των χρωμάτων, για την σχεδίαση των διαφορετικών τύπων γεωμετρικών σχημάτων όπως Point, LineString, Polygon, MultiPolygon καθώς και για την εμφάνιση μηνυμάτων με την μορφή “InfoWindow”. Επιπλέον εδώ προσθέτουμε δύο “Listener” τους “zoom\_changed” που μας ενημερώνει για την τρέχουσα κλίμακα του χάρτη και “click” που μας επιστρέφει το σημείο που ο χρήστης έκανε κλικ με το ποντίκι ή στην οθόνη αφής. Ακολουθεί το σχετικό κομμάτι javascript:

```
function afterupdatehtml(sender)
{
    //Δημιουργία αντικειμένου που θα περιέχει τις πληροφορίες επιπέδων
    FCLayers=new Object();
    FCLayers.FeatureCollection=new Array();
    //Σχεδιασμός του Pin για τον σχεδιασμό σημείων
    FCLayers.pinSymbol=(function(color) {
        return {
            path: 'M0 0 L3 -10 L-3 -10 Z',
            fillColor: color,
            fillOpacity: 1,
            strokeColor: '#333',
            strokeWeight: 1,
            scale: 1,
        }
    });
    //Function προσθήκης επιπέδου
    FCLayers.addFC=(function(aFC,aFCcolor) {
        var aFCJson=JSON.parse(aFC);
        if (!FCLayers.FeatureCollection) FCLayers.FeatureCollection=new Array();
        FCLayers.FeatureCollection[FCLayers.FeatureCollection.length]=aFCJson;
        googleMap.data.addGeoJson(aFCJson);
        return FCLayers.FeatureCollection.length-1;
    });
    //Function διαγραφής επιπέδου με id
    FCLayers.delFCbyId=(function(alayer) {
        var aFCidx;
        googleMap.data.forEach(function (feature,index) {
            if (feature.getProperty('layer')==alayer) {
                aFCidx=index;
                googleMap.data.remove(feature);
            }
        });
        if (aFCidx)
            FCLayers.FeatureCollection.splice(aFCidx,1);
    });
    //Function διαγραφής επιπέδου
    FCLayers.delFC=(function(aFCidx) {
        var alayer=FCLayers.FeatureCollection[aFCidx].layer;
        googleMap.data.forEach(function (feature) {
            if (feature.layer==alayer)
                googleMap.data.remove(feature);
        });
        FCLayers.FeatureCollection.splice(aFCidx,1);
    });
    //Function διαγραφής όλων των επιπέδων
    FCLayers.clearLayers=(function() {
        googleMap.data.forEach(function (feature) {
            googleMap.data.remove(feature);
        });
        FCLayers.FeatureCollection.splice(0,FCLayers.FeatureCollection.length-1);
    });
}
```

```
//Μετατροπή χρώματος HEX σε Luminance
FCLayers.ColorLuminance=(function(hex, lum) {
    hex = String(hex).replace(/^[^0-9a-f]/gi, '');
    if (hex.length < 6) {
        hex = hex[0]+hex[0]+hex[1]+hex[1]+hex[2]+hex[2];
    }
    lum = lum || 0;
    var rgb = "#", c, i;
    for (i = 0; i < 3; i++) {
        c = parseInt(hex.substr(i*2,2), 16);
        c = Math.round(Math.min(Math.max(0, c + (c * lum)),
255)).toString(16);
        rgb += ("00"+c).substr(c.length);
    }
    return rgb;
});
//Δημιουργία προεπιλεγμένου σημείου για το κέντρο του χάρτη
var latlng = new google.maps.LatLng(38.555, 19.6555);
var myOptions = {
    zoom: 7,
    center: latlng,
    mapTypeId: google.maps.MapTypeId.ROADMAP
};

//Σύνδεση χάρτη με το "div"
var umap = document.getElementById("uni_map_canvas");
var map = new google.maps.Map(umap, myOptions);
googleMap = map;

//Ορισμός στυλ του χάρτη
googleMap.data.setStyle(function(feature) {
    //Point
    if (feature.getGeometry().getType()=== 'Point')
        return {
            icon: FCLayers.pinSymbol(feature.getProperty('color')) ,
        };
    //MultiLineString
    if (feature.getGeometry().getType()=== 'MultiLineString')
        return {
            strokeColor: feature.getProperty('color') ,
            strokeOpacity: 0.7,
            strokeWeight: 2,
        };
    //MultiPolygon
    if (feature.getGeometry().getType()=== 'MultiPolygon')
        return {
            fillColor: feature.getProperty('color') ,
            fillOpacity: 0.7,
            strokeColor: FCLayers.ColorLuminance(feature.getProperty('color'), -
0.3) ,
            strokeWeight: 1,
        };
    //Polygon
    if (feature.getGeometry().getType()=== 'Polygon')
        return {
            fillColor: feature.getProperty('color') ,
            fillOpacity: 0.7,
            strokeColor: FCLayers.ColorLuminance(feature.getProperty('color'), -
0.3) ,
            strokeWeight: 1,
        };
    //LineString
    if (feature.getGeometry().getType()=== 'LineString')
        return {
```



```
strokeColor: feature.getProperty('color') ,
strokeOpacity: 0.7,
strokeWeight: 2,
});
});

//Παράθυρο popup όπου θα εμφανίζονται πληροφορίες
infobox = new google.maps.InfoWindow;

//Εδώ ορίζουμε την συμπεριφορά του κλικ, εμφανίζει το παράθυρο πληροφοριών
map.data.addListener('click', function(event) {
    var myHTML = event.feature.getProperty('name');
    infobox.setContent("<div style='width:150px; text-align:
center;'>"+myHTML+"</div>");
    infobox.setPosition(event.latLng);
    infobox.setOptions({pixelOffset: new google.maps.Size(0,0)});
    infobox.open(map);
});

//Callback, ενημερώνει την εφαρμογή μας ότι άλλαξε το zoom
google.maps.event.addListener(map, 'zoom_changed',
function() {
    MapsDesignFrm.UniTrackBar1.SetValue(this.getZoom());
}
);

//Callback, ενημερώνει την εφαρμογή μας για το σημείο του κλικ
google.maps.event.addListener(map, 'click',
function(e) {
    ajaxRequest(MapsDesignFrm.UniHTMLFrame1,
        'mapClick',
        ['lat='+e.latLng.lat(), 'lng='+e.latLng.lng()]);
    }
);
}
```

### Εκτέλεση Query

Κάθε φορά που ο χρήστης πατά το πλήκτρο εκτέλεση, η εφαρμογή επεξεργάζεται το sql statement και αναζητεί αν υπάρχουν macros (παράμετροι χρήστη, αρχίζουν με "!"). Αν υπάρχουν τα αντικαθιστά με τις προεπιλεγμένες τους τιμές. Η εκτέλεση του query πραγματοποιείται με το παρακάτω:

```
procedure TMapsDesignFrm.UniButton1Click(Sender: TObject);
var i:integer;
begin
    //Αποθήκευση ερωτήματος χρήστη στον πίνακα wcqueries
    if wcqueries.sql.AsWideString<>UniSyntaxEdit1.Lines.Text then begin
        wcqueries.Edit;
        wcqueries.sql.AsWideString:=UniSyntaxEdit1.Lines.Text;
    end;
    if (wcqueries.State=dsEdit) or (wcqueries.State=dsInsert) then
        wcqueries.Post;

    //Κλείσιμο του προηγούμενου dataset, σε περίπτωση που ήταν ανοικτό
    FDQuery.Active:=false;
    //Εισαγωγή νέου sql στο dataset
    FDQuery.SQL.Text:=StringReplace(wcqueries.sql.AsString,#13#10,'
',[rfReplaceAll]) ;
    //Απενεργοποίηση των συνδεδεμένων control, για γρηγορότερη εκτέλεση
    wcparams.DisableControls;
```



```
//Αναζήτηση macros και ενημέρωση με νέες τιμές
for I := 0 to FDQuery.MacroCount-1 do begin
    if
        not
        wparams.Locate('paramname',FDQuery.Macros[i].Name,[loCaseInsensitive]) then
        raise Exception.Create('Η παράμετρος '+FDQuery.Macros[i].Name+' δεν είναι
        ορισμένη!');
        if wparams.defaultvalue.AsWideString='' then

FDQuery.Macros[i].Value:=StringReplace(StringReplace(wparams.whererepart.AsWideSt
ring,#13#10,'',[rfReplaceAll]),'@','Null',[rfReplaceAll])
        else

FDQuery.Macros[i].Value:=StringReplace(StringReplace(wparams.whererepart.AsWideSt
ring,#13#10,'',[rfReplaceAll]),'@',wparams.defaultvalue.AsWideString,[rfReplace
All]);
    end;

//Ενεργοποίηση συνδεδεμένων controls
wparams.EnableControls;
//Άνοιγμα του dataset, εκτέλεση του query στην βάση
FDQuery.Active:=true;

//Σε περίπτωση που υπάρχουν Memo πεδία (τύπος BLOB), κατευθύνουμε την
//εφαρμογή να επιστρέψει στο grid μόνο τους 512 χαρακτήρες, για ταχύτητα.
//Αυτό το αναλαμβάνει το onGetText event χρησιμοποιώντας την "getMemoText"
for i:=0 to fdquery.fields.Count-1 do
    if FDQuery.Fields[i].DataType=ftWideMemo then
        FDQuery.Fields[i].OnGetText:=getMemoText;

end;
```

#### Εμφάνιση Γεωγραφικού Ερωτήματος

Κάθε φορά που ο χρήστης επιθυμεί να εμφανίσει το ερώτημά του στον χάρτη εκτελείται το παρακάτω κομμάτι κώδικα:

```
procedure TMapsDesignFrm.UniToolButton3Click(Sender: TObject);
var str,acolor:widestring;
    i:integer;
    captionfield:Tfield;
    sl:Tstringstream;
begin
    //Αν το dataset είναι κλειστό, το πρόγραμμα επιστρέφει
    if not FDQuery.Active then
        exit;
    //Έλεγχος αν υπάρχει γεωγραφική στήλη
    if wqueries.geomcolumn.IsNull then begin
        ShowMessage('Προσοχή δεν έχετε συμπληρώσει το πεδίο "Γεωγραφική στήλη"!');
        exit;
    end;
    //Δημιουργία Αλφαριθμητικού stream, όπου συγκεντρώνουμε τα δεδομένα μας
    sl:=Tstringstream.Create;
    //Απενεργοποίηση controls για ταχύτητα
    FDQuery.DisableControls;
    //Εύρεση του χρώματος εμφάνισης
    acolor:=wqueries.color.AsWideString;
    if acolor='' then
        acolor:='#f00';
    //Πεδίο που περιέχει το κείμενο που θα εμφανίζει το InfoBox
    if wqueries.descrcolumn.IsNull then
        captionfield:=FDQuery.FieldByName(wqueries.geomcolumn.AsWideString)
    else
```



```
captionfield:=FDQuery.FieldByName(wcqueriesdescrcolumn.AsWideString);

i:=FDQuery.FieldList.IndexOf(wcqueriesgeomcolumn.AsString);
//Δημιουργία αντικειμένου JSON για να το στείλουμε στην σελίδα
sl.WriteString('{"type": "FeatureCollection","properties": {"name":
"'+wcqueriescaption.AsWideString+'"},"layer": "'+wcqueriesid.AsWideString+'", "fe
atures": [');
FDQuery.First;
//Ανατρέχουμε σε όλες τις γραμμές του dataset και προσθέτουμε την γεωμετρία
//και τα δεδομένα τις
while not FDQuery.Eof do begin
  if FDQuery.Bof then
    sl.WriteString('{"type":
"Feature","layer": "'+wcqueriesid.AsWideString+'", "geometry": "'+FDQuery.fields[i]
.AsString+'", "properties": {"name":
"'+StringReplace(captionfield.AsWideString, ' ', '\ ', [rfReplaceAll])+'", "color
": "'+acolor+'", "layer": "'+wcqueriesid.AsWideString+'"}')
  else
    sl.WriteString('{"type":
"Feature","layer": "'+wcqueriesid.AsWideString+'", "geometry": "'+FDQuery.fields[i]
.AsString+'", "properties": {"name":
"'+StringReplace(captionfield.AsWideString, ' ', '\ ', [rfReplaceAll])+'", "color
": "'+acolor+'", "layer": "'+wcqueriesid.AsWideString+'"}');
  FDQuery.Next;
end;
//Ολοκλήρωση αντικειμένου JSON
sl.WriteString(']');
//Αποστολή στην σελίδα. Πρώτα γίνεται κλήση στο delFCbyId για να σβήσει
//το προηγούμενο επίπεδο. Μετά καλούμε το addFC και προσθέτουμε το νέο

UniSession.JSCode('FCLayers.delFCbyId(''+wcqueriesid.AsWideString+'');FCLayers.ad
dFC(''+sl.DataString+'', '#ddd');');//+
//Αποδέσμευση μνήμης stream
sl.Free;
//Επαναφορά dataset στην πρώτη γραμμή
fdquery.First;
FDQuery.EnableControls;
end;
```

## 4.2. MapsViewer

Το MapsViewer είναι μια υποστηρικτική εφαρμογή που συνδυάζεται με το “Maps” και σαν στόχο της έχει την προβολή διαδραστικών χαρτών εύκολα, χωρίς την ανάγκη προγραμματισμού. Η ανάπτυξη της εφαρμογής έγινε επίσης χρησιμοποιώντας το Embarcadero Delphi με την επέκταση uniGUI. Για τους χάρτες επιλέχθηκε το Google maps API.

Το project MapsViewer έχει υλοποιηθεί με την χρήση του uniGUI με τις βιβλιοθήκες Sencha Touch, το οποίο είναι προσανατολισμένο στις φορητές συσκευές με οθόνη αφής. Τα controls είναι μεγαλύτερα και υποστηρίζονται λειτουργίες gesture.

Όμοια με το προηγούμενο Project, υπάρχει το ServerModule με τις ίδιες ρυθμίσεις

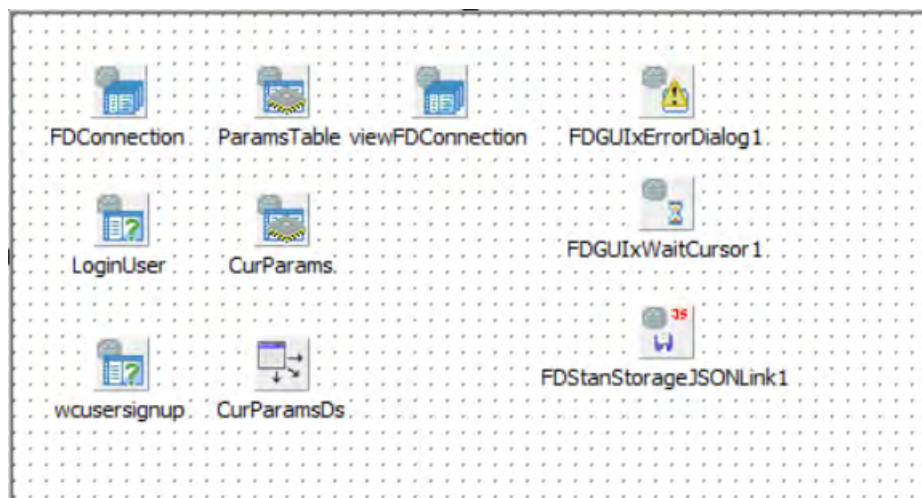
Παρακάτω περιγράφουμε τα Unit του project:

❖ ServerModule.pas

Είναι το module του server μας και έχει την ίδια λειτουργικότητα με το “Maps” με την διαφορά ότι του έχουμε ορίσει σαν προεπιλεγμένη Port την 8079. Με αυτό τον τρόπο δημιουργήσαμε ένα δεύτερο web server που εξυπηρετεί σε αυτή την θύρα.

#### ❖ MainModule.pas

Αυτό το module δημιουργείται για κάθε ένα session του web server. Η λειτουργία του είναι προσανατολισμένη στην επικοινωνία με την βάση δεδομένων όπως και στο “Maps”. Στο τρέχων module χρησιμοποιείται για να συνδεθεί στο schema “postgres”, όπου θα πραγματοποιηθεί το authentication και στο schema του χρήστη απ’ όπου θα αντλήσει δεδομένα των γεωγραφικών ερωτημάτων.



Εικόνα 4.2.1 Σχεδίαση του UniMainModule

Το αντικείμενο “UniMainModule” είναι ίδιο με αυτό του Project “Maps” με επιπλέον το συστατικό “viewFDConnection” που πραγματοποιεί την σύνδεση με το schema του χρήστη. Για την δεύτερη σύνδεση έχουμε υλοποιήσει το “viewGetQuery” που εκτελεί την ίδια λειτουργία με το “GetQuery”, αλλά για λογαριασμό της επιπλέον σύνδεσης.

#### ❖ Mainm.pas

Στη unit “Mainm” αναπτύξαμε την λειτουργία του διαδραστικού χάρτη. Στο πάνω μέρος δημιουργήσαμε την εργαλειοθήκη ελέγχου του χάρτη, στο κέντρο το συστατικό “UnimHTMLFrame1” που εμφανίζεται ο χάρτης και στο κάτω μέρος υπάρχει ένα slider που διαμορφώνει την κλίμακα του χάρτη.

Ο τρόπος που υλοποιήθηκε ο χάρτης είναι όμοιος με το “Maps”. Στην περίπτωση του “Mainm” προσθέσαμε τη δυνατότητα παραμετροποίησης της προβολής μέσω των παραμέτρων κλήσης του Url. Ανάλογα με τις παραμέτρους αυτές καθορίζουμε την κλίμακα, το κέντρο του χάρτη καθώς και το ποια controls θα εμφανίζονται. Ο σχετικός κώδικας του Event “onFormCreate” εμφανίζεται παρακάτω:

```
procedure TMainmForm.UnimFormCreate(Sender: TObject);  
var id,i:Integer;  
u,p,r,dbuser:widestring;
```



```
begin
//Διάβασμα παραμέτρων του Url
u:=UniSession.Query['user'];
p:=UniSession.Query['token'];
r:=UniSession.Query['report'];
strlat:=UniSession.Query['lat'];
strlng:=UniSession.Query['lng'];
strzoom:=UniSession.Query['zoom'];
toolbar:=UniSession.Query['toolbar'];
zoomslider:=UniSession.Query['zoomslider'];
if toolbar='0' then begin
    UnimToolBar1.Visible:=false;
    UnimToolBar1.Height:=0;
end;
if zoomslider='0' then begin
    UnimPanel3.Visible:=false;
    UnimPanel3.Height:=0;
end;

//Έλεγχος στοιχείων του χρήστη
id:=UniMainModule.getQueryInteger('select      id      from      wcuser      where
upper(email)=upper(:1) and upper(password)=upper(:2)',vararrayof([u,p]));
if id=0 then begin
    ShowMessage('Λάθος στοιχεία σύνδεσης!');
    exit;
end
else
    Caption:='Ok';
//Σύνδεση στην βάση "postgres"
UniMainModule.FDConnection.Connected:=true;
UniMainModule.execQuery('delete      from      wcloggedin      where      usrid=:1      and
endpoint='webviewer'',id);
UniMainModule.execQuery('insert      into      wcloggedin      (usrid,sessionid,endpoint)
values(:1,:2,'webviewer')',vararrayof([id,UniSession.SessionId]));
checklogin();

dbuser:='user_'+UniMainModule.CurParamsemail.AsWideString;
UniMainModule.viewFDConnection.Connected:=false;

//Σύνδεση στην schema χρήστη
UniMainModule.viewFDConnection.Params.Text:='Database=postgres'+#13#10+
        'User_Name='+dbuser+#13#10+
'Password='+UniMainModule.CurParamshashedpassword.AsWideString+#13#10+
        'CharacterSet=UTF8'+#13#10+
        'UnknownFormat=BYTEA'+#13#10+
        'ApplicationName=GIS Maps'+#13#10+
        'Server=127.0.0.1'+#13#10+
        'DriverID=PG';
UniMainModule.viewFDConnection.Connected:=true;

//Εύρεση τίτλου προβολής
UnimLabel1.Caption:=UniMainModule.viewGetQueryString('select      caption      from
wcqueries where linenum=:1',strtoint(r));

wcqueries.Active:=true;
wcparams.Active:=true;
FDQuery.Active:=false;

//Εύρεση γραμμής γεωγραφικού ερωτήματος
if wcqueries.Locate('linenum',strtoint(r),[]) then begin
    FDQuery.SQL.Text:=wcqueries.sql.AsWideString;
    wcparams.DisableControls;
    //Αρχικοποίηση macros
```

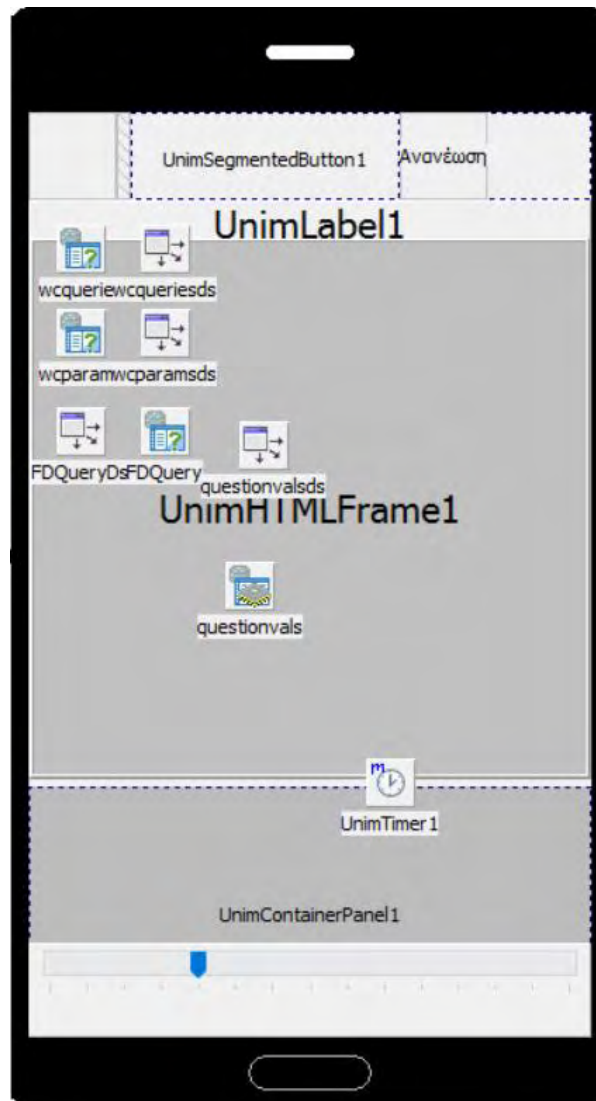
```

for I := 0 to FDQuery.MacroCount-1 do begin
  if
    wparams.Locate('paramname',FDQuery.Macros[i].Name,[loCaseInsensitive]) then      not
    raise Exception.Create('Η παράμετρος '+FDQuery.Macros[i].Name+' δεν είναι
    ορισμένη!');
    if wparams.defaultvalue.AsWideString='' then

    FDQuery.Macros[i].Value:=StringReplace(wparams.whererepart.AsWideString,'@','Null',[
    rfReplaceAll])
    else

    FDQuery.Macros[i].Value:=StringReplace(wparams.whererepart.AsWideString,'@',wparams
    defaultvalue.AsWideString,[rfReplaceAll]);
    end;
  end
  else begin
    ShowMessage('Δεν βρέθηκε το ερώτημα!');
    exit;
  end;
end;
end;

```



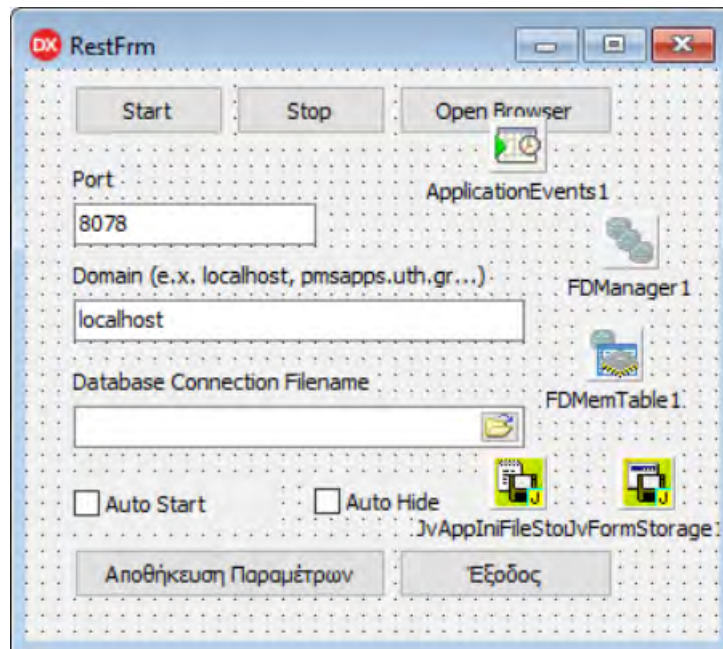
Εικόνα 4.2.2 Σχεδίαση του διαδραστικού χάρτη, φόρμα MainForm

### 4.3. RestService

Στο “RestService” έχουμε υλοποιήσει την απαραίτητη υπηρεσία REST API. Σε αυτό το Project έχουμε δημιουργήσει δύο units τα “Restform” και “RestDM”. Το πρώτο το χρησιμοποιούμε για να παραμετροποιήσουμε την υπηρεσία και το δεύτερο μας δίνει πρόσβαση στο schema “postgres” και στο schema του χρήστη.

#### Restform

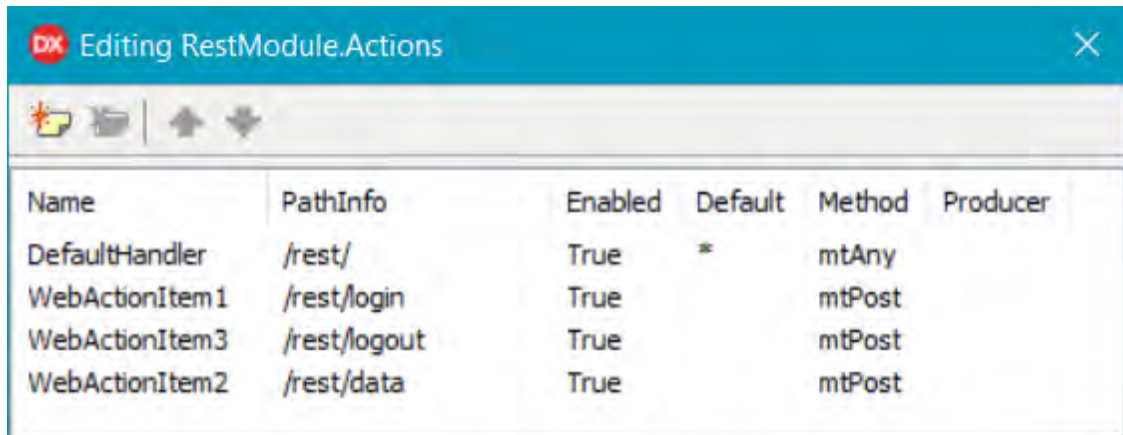
Με την RestForm μπορούμε να ορίσουμε το Port που θα εξυπηρετεί το REST API, το όνομα του server, το αρχείο σύνδεσης της βάσης δεδομένων, καθώς επίσης και το αν θα ξεκινά αυτόματα η υπηρεσία ή όχι.



Εικόνα 4.3.1 Σχεδίαση φόρμας RestFrm

#### RestDM

Αυτή η unit είναι υπεύθυνη για την λειτουργία του Rest API. Το unit περιέχει το αντικείμενο RestModule τύπου TRestModule και μας δίνει την απαραίτητη λειτουργικότητα, μέσω των χαρακτηριστικών action που παρέχει, για να σχεδιάσουμε πάνω του API. Επιπλέον περιέχει δύο συνδέσεις μια “FDConnection” για το schema “postgres” και μια “FDConnectionRest” για το schema του χρήστη. Το “InsertQuery” χρησιμοποιείται για την εισαγωγή δεδομένων και θα εξηγήσουμε τον τρόπο με τον οποίο το καταφέρνει αναλυτικότερα παρακάτω. Τα actions που υλοποιήσαμε είναι τα :



Name	PathInfo	Enabled	Default	Method	Producer
DefaultHandler	/rest/	True	*	mtAny	
WebActionItem1	/rest/login	True		mtPost	
WebActionItem3	/rest/logout	True		mtPost	
WebActionItem2	/rest/data	True		mtPost	

Εικόνα 4.3.3 Actions του Rest API

#### 1. Default Handler “/”

Επιστρέφει ένα μήνυμα με τον τίτλο της υπηρεσίας :

```
procedure TRestModule.WebModule1DefaultHandlerAction(Sender: TObject;
  Request: TWebRequest; Response: TWebResponse; var Handled: Boolean);
begin
  Response.Content :=
    '<html>' +
    '<head><title>GISMaps Rest Server</title></head>' +
    '<body><h2>GISMaps Rest Server</h2>Web Server Application</body>' +
    '</html>';
end;
```

#### 2. Login

Πραγματοποιεί το login, είναι προαπαιτούμενο για οποιαδήποτε επόμενη κλήση:

```
procedure TRestModule.WebModule1WebActionItem1Action(Sender: TObject;
  Request: TWebRequest; Response: TWebResponse; var Handled: Boolean);
var dbuser,dbpassword,email,password,c,sessionid,h:widestring;
    id:Integer;
    v:variant;
    td:tdataset;
begin
  //Parsing του body του request σε ένα αντικείμενο JSON
  v := _Json(Request.Content);
  email:=v.email;
  password:=v.password;
  //Σύνδεση στο schema "postgres"
  if not FDConnection.Connected then
    FDConnection.Connected:=true;
  //Έλεγχος στοιχείων σύνδεσης
  td:=getQuery('select id,email,password from wcuser where email=:1 and password=:2
and isactive=1',
    varrayof([LowerCase(email),hashString(password)]));
  id:=td.FieldByName('id').AsInteger;
  dbuser:='user_'+td.FieldByName('email').AsWideString;
  dbpassword:=td.FieldByName('password').AsWideString;
  td.Free;
  //Σε περίπτωση λάθους στοιχείων επιστρέφουμε το παρακάτω
  if id=0 then begin
    Response.Content:='application/json';
    Response.Content:='{msg : "credential error"}';
    exit;
  end;
```





```
//Αποσύνδεση εκκρεμών session, διαγραφή από τον αντίστοιχο πίνακα
execQuery('delete from wcloggedin where usrid=:1 and endpoint=''rest'',id);
sessionid:=hashString(inttostr(id)+inttostr(GetTickCount));
//Σύνδεση session, εισαγωγή στον αντίστοιχο πίνακα
execQuery('insert into wcloggedin (usrid,sessionid,endpoint)
values(:1,:2,''rest'')',
vararrayof([id,SessionId]));
//Εισαγωγή στοιχείων σύνδεσης στο schema χρήστη
FDConnectionRest.Params.Text:='Database=postgres'+#13#10+
'User_Name='+dbuser+#13#10+
'Password='+dbpassword+#13#10+
'CharacterSet=UTF8'+#13#10+
'UnknownFormat=BYTEA'+#13#10+
'ApplicationName=GIS Maps Rest Server'+#13#10+
'Server=127.0.0.1'+#13#10+
'DriverID=PG';
//Επιστρεφόμενο αποτέλεσμα κλήσης, επιστρέφεται το sessiontoken
Response.ContentType:='application/json';
Response.Content:='{ "msg" : "Loggedin", "sessiontoken" : "' +sessionid+'"}';
end;
```

### 3. Logout

Αποσύνδεση και αποδέσμευση του sessiontoken:

```
procedure TRestModule.RestModuleWebActionItem3Action(Sender: TObject;
Request: TWebRequest; Response: TWebResponse; var Handled: Boolean);
var sessionid:widestring;
v:variant;
begin
//Parsing του body του request σε ένα αντικείμενο JSON
v := _Json(Request.Content);
sessionid:=v.sessiontoken;
if not FDConnection.Connected then
FDConnection.Connected:=true;
//Διαγραφή από πίνακα συνδέσεων
execQuery('delete from wcloggedin where sessionid=:1',sessionid);
//Επιστροφή μηνύματος αποσύνδεσης
Response.ContentType:='application/json';
Response.Content:='{ "msg" : "Logged out"}';
end;
```

### 4. Data

Όλες οι υπόλοιπες λειτουργίες υλοποιούνται από το “Data” :

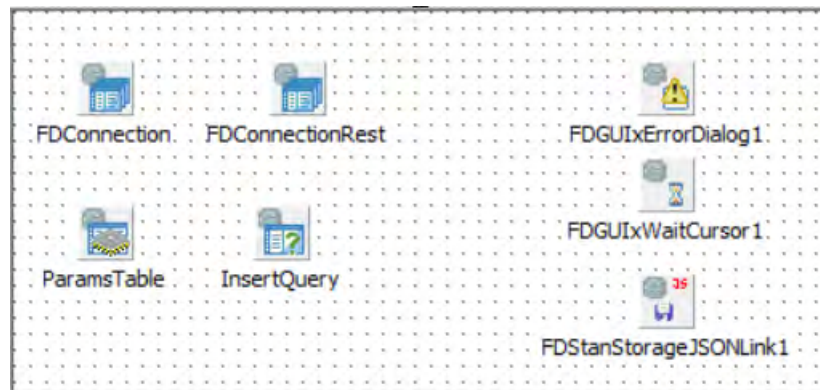
```
procedure TRestModule.RestModuleWebActionItem2Action(Sender: TObject;
Request: TWebRequest; Response: TWebResponse; var Handled: Boolean);
var dbuser,table,sessionid,w,o,f,action,sql,p,fieldname:widestring;
td:tdataset;
v,hv,tabldata,tablerow,prms,resultfields,dt:variant;
ss:TStringstream;
rowsaffected,i,j,idx:integer;
h:UTF8String;
fs:TFormatSettings;
tft:Tfieldtype;
begin
//Ορισμός προεπιλεγμένης μορφής ημερομηνία ώρας και αριθμών
fs.DateSeparator:='-';
fs.TimeSeparator:=':';
fs.ShortDateFormat:='yyyy-mm-dd HH:nn:ss.zzz';
fs.DecimalSeparator:',';
fs.ThousandSeparator:='.';
System.SysUtils.FormatSettings := fs;
```



```
//Parsing του body του request σε ένα αντικείμενο JSON
v := _Json(Request.Content);
//Έλεγχος ορθής σύνδεσης και sessiontoken
sessionid:='';
try
hv:=v.sessiontoken;
except
Response.Content:='{ "msg":"error", "info":"Invalid session.."}';
exit;
end;
if (not VarIsEmpty(hv)) and (not VarIsNull(hv)) then
sessionid:=hv
else begin
Response.Content:='{ "msg":"Not Logged in.."}';
exit;
end;
//Ανάγνωση παραμέτρων του request body
hv:=v.table;
if (not VarIsEmpty(hv)) and (not VarIsNull(hv)) then
table:=hv;
action:='';
hv:=v.action;
if (not VarIsEmpty(hv)) and (not VarIsNull(hv)) then
action:=hv;
hv:=v.sql;
sql:='';
if (not VarIsEmpty(hv)) and (not VarIsNull(hv)) then
sql:=hv;
hv:=v.params;
if (not VarIsEmpty(hv)) and (not VarIsNull(hv)) then
p:=hv;
hv:=v.fields;
f:='';
if (not VarIsEmpty(hv)) and (not VarIsNull(hv)) then
f:=hv;
if f='' then
f:='*';
hv:=v.where;
if (not VarIsEmpty(hv)) and (not VarIsNull(hv)) then
w:=hv;
if w<>'' then
w:='where '+w;
hv:=v.orderby;
if (not VarIsEmpty(hv)) and (not VarIsNull(hv)) then
o:=hv;
if o<>'' then
o:='order by '+o;
if not FDConnectionRest.Connected then
FDConnectionRest.Connected:=true;

//Έλεγχος αν ο χρήστης έχει κάνει login
dbuser:=getDBUserName(sessionid);
if dbuser='' then begin
Response.Content:='{msg : "Not logged in!"}';
exit;
end;
//Εξήγηση κλήσης δεδομένων πίνακα
if action='select' then begin
if table<>'' then
td:=userGetQuery('select '+f+' from '+table+' '+w+' '+o,'')
else
td:=userGetQueryDocVariant(sql,p);
tabledata:=DataSetToJSON(td);
```

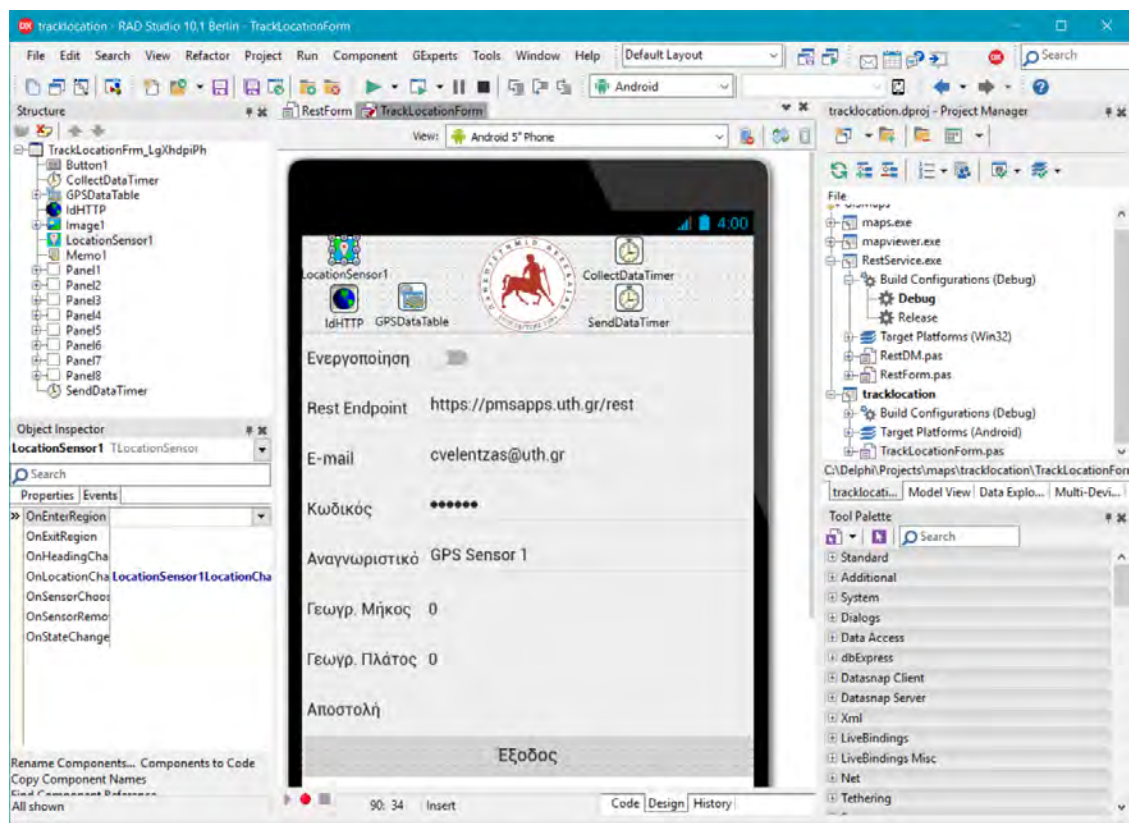
```
Response.Content:='{ "msg": "Ok",  
"table": "' + StringReplace(table, '"', '\'', [rfReplaceAll]) + '",  
"rows": '+inttostr(td.RecordCount)+', "data": '+tabledata+'}';  
td.Free;  
end;  
//Εξήγηση κλήσης εκτέλεσης DML, DDL statment  
if action='execute' then begin  
    rowsaffected:=userExecQueryDocVariant(sql,p);  
    Response.Content:='{ "msg": "Ok", "rowsaffected": '+inttostr(rowsaffected)+'}';  
end;  
//Εισαγωγή δεδομένων σε πίνακα της βάσης  
if action='insert' then begin  
    hv:=v.data;  
    if (not VarIsEmpty(hv)) and (not VarIsNull(hv)) then  
        tabledata:=_Json(hv)  
    else begin  
        Response.Content:='{ "msg": "Data missing!"}';  
        exit;  
    end;  
    InsertQuery.Active:=false;  
    InsertQuery.Command.CommandKind:=skInsert;  
    InsertQuery.SQL.Text:='select * from '+table+' where 0=1';  
    InsertQuery.Active:=true;  
    for j := 0 to tabledata._Count-1 do begin  
        InsertQuery.Append;  
        tablerow:=tabledata._(j);  
        for i := 0 to tablerow._Count-1 do begin  
            fieldname:=DocVariantData(tablerow).Names[i];  
            if InsertQuery.FieldList.IndexOf(fieldname)>=0 then begin  
                dt:= DocVariantData(tablerow).Values[i];  
                tft:=InsertQuery.FieldByName(fieldname).DataType;  
                tft:=ftInteger;  
                try  
                    if InsertQuery.FieldByName(fieldname).DataType=ftTimeStamp then  
                        InsertQuery.FieldByName(fieldname).value:=strtodatetime(dt)  
                    else if InsertQuery.FieldByName(fieldname).DataType=ftDate then  
                        InsertQuery.FieldByName(fieldname).value:=strtodate(dt)  
                    else if InsertQuery.FieldByName(fieldname).DataType=ftDateTime then  
                        InsertQuery.FieldByName(fieldname).value:=strtodatetime(dt)  
                    else if InsertQuery.FieldByName(fieldname).DataType=ftFloat then  
                        InsertQuery.FieldByName(fieldname).value:=  
StrToFloat(StringReplace(dt, '.', ',', [], []))  
                    else  
                        InsertQuery.FieldByName(fieldname).AsVariant:=dt;  
                except  
                    on e:exception do begin  
                        Response.Content:='{ "msg": "Exception : '+e.Message+'}';  
                        InsertQuery.CancelUpdates;  
                        exit;  
                    end;  
                end;  
            end;  
        end;  
    end;  
    InsertQuery.ApplyUpdates();  
    //Επιστροφή δεδομένων πίνακα που μόλις εισάγαμε  
    tabledata:=DataSetToJSON2(InsertQuery);  
    Response.Content:='{ "msg": "Ok",  
"table": "' + StringReplace(table, '"', '\'', [rfReplaceAll]) + '",  
"rows": '+inttostr(InsertQuery.RecordCount)+', "data": '+tabledata+'}';  
    InsertQuery.Active:=false;  
end;  
end;
```



Εικόνα 4.3.2 Σχεδίαση RestModule

#### 4.4. TrackLocation

Το “TrackLocation” μας βοηθά να συλλέξουμε δεδομένα στίγματος και να τα στείλει στο REST service. Η ανάπτυξή του έγινε με το Embarcadero Delphi και χρησιμοποιήθηκε η δυνατότητα “Multi-Device-Application”, η οποία με κοινή βάση κώδικα δημιουργεί εφαρμογές για τα λειτουργικά συστήματα Windows, Android, OSX, iOS και τις αντίστοιχες συσκευές τους.



Εικόνα 4.4.1 Ανάπτυξη “Multi-Device-Project”

#### Εύρεση στίγματος



Για την ανάγνωση του GPS χρησιμοποιήθηκε το συστατικό “LocationSensor1” το οποίο μας ενημερώνει για το στίγμα με το Event onLocationChanged:

```
procedure TTrackLocationFrm.LocationSensor1LocationChanged(Sender: TObject;  
  const OldLocation, NewLocation: TLocationCoord2D);  
var  
  URLString: String;  
  LSettings: TFormatSettings;  
begin  
  //Ανάγνωση μορφής αριθμών, ημερομηνίας κτλ  
  LSettings := FormatSettings;  
  //Ορισμός χαρακτήρα δεκαδικών  
  FormatSettings.DecimalSeparator := '.';  
  //Εμφάνιση γεωγραφικού μήκους και πλάτους  
  Label6.Text := Format('%2.6f', [NewLocation.Latitude]);  
  Label8.Text := Format('%2.6f', [NewLocation.Longitude]);  
  //Αποθήκευση γεωγραφικού μήκους και πλάτους στις παρακάτω μεταβλητές  
  curLat:=NewLocation.Latitude;  
  curLng:=NewLocation.Longitude;  
end;
```

#### Αποθήκευση στίγματος

Κάθε λεπτό αποθηκεύουμε το τρέχον στίγμα σε ένα προσωρινό πίνακα, τον GPSTable. Τα πεδία που αποθηκεύουμε είναι τα:

- ❖ Name  
Το όνομα που έχουμε εισάγει
- ❖ Lat  
Το γεωγραφικό πλάτος
- ❖ Lng  
Το γεωγραφικό μήκος
- ❖ Trntime  
Ημερομηνία και ώρα καταγραφής στίγματος
- ❖ Issent  
Αν έχει αποσταλεί το στίγμα τότε ενημερώνουμε με «1», διαφορετικά με «0»
- ❖ Ispacked  
Αν έχει προετοιμασθεί για αποστολή το ενημερώνουμε με «1», διαφορετικά με «0»

Η διαδικασία αποθήκευσης ξεκινά με το Event OnTimer του αντικείμενου “CollectDataTimer”:

```
procedure TTrackLocationFrm.CollectDataTimerTimer(Sender: TObject);  
begin  
  //Αν δεν υπάρχει στίγμα, εμφανίζει μήνυμα και επιστρέφει  
  if abs(curLat+curLng)<0.0001 then begin  
    label10.Text:='Δεν υπάρχει σήμα GPS!';  
    exit;  
  end;  
  //Μετακινούμαστε στην τελευταία εγγραφή του πίνακα και εισάγουμε μια νέα εγγραφή  
  GPSTable.Last;  
  GPSTable.Insert;  
  GPSTableName.AsWideString:=edit4.Text;
```



```
GPSDataTable.TrnTime.AsDateTime:=now();  
GPSDataTable.Lat.AsFloat:=curLat;  
GPSDataTable.Lng.AsFloat:=curLng;  
GPSDataTable.CheckBrowseMode;  
end;
```

### Αποστολή

Η αποστολή δεδομένων πραγματοποιείται κάθε 10 λεπτά. Η διαδικασία ξεκινά με το Event OnTimer του timer “SendDataTimer”:

```
procedure TTrackLocationFrm.SendDataTimer(Sender: TObject);  
var resp, sessiontoken, msg, dtstr, r: string;  
    ssRequest, ssResponse: TStringStream;  
    v: Variant;  
    sl: TStringList;  
begin  
    //Αν δεν υπάρχουν αποθηκευμένα στίγματα, επιστροφή  
    if GPSDataTable.RecordCount=0 then exit;  
  
    //Διαδικασία Login  
    sessiontoken:='';  
    ssRequest:=TStringStream.Create('{"email":           "'+edit2.Text+'", "password":  
    "'+edit3.Text+'"}');  
    ssResponse:=TStringStream.Create;  
    try  
        IdHTTP.Post( edit1.Text+'/login', ssRequest, ssResponse);  
    except  
        on e:exception do begin  
            //Αν δεν επιτύχει η κλήση για login τότε εμφανίζουμε το μήνυμα αποτυχίας  
            Label10.Text:=e.Message;  
            ssRequest.Free;  
            ssResponse.Free;  
        end;  
    end;  
    //Ελεγχος για επιτυχημένο login  
    try  
        v:= JSONVariant(ssResponse.DataString);  
        msg:=v.msg;  
        if msg='Loggedin' then  
            sessiontoken:=v.sessiontoken  
        else begin  
            Label10.Text:='Αποτυχία αποστολής :'+msg;  
            ssRequest.Free;  
            ssResponse.Free;  
            exit;  
        end;  
    except  
    end;  
    //Ελευθέρωση μνήμης  
    ssRequest.Free;  
    ssResponse.Free;  
    //Αν δεν λάβαμε sessiontoken, επιστροφή  
    if sessiontoken='' then exit;  
  
    //Αποστολή δεδομένων, δημιουργία JSON  
    dtstr:='{"action":      "insert", "sessiontoken":      "'+sessiontoken+'",      "table":  
    "\"user_'+edit2.Text+'\".\"gpsdata\"", "data": [';  
    //Απενεργοποίηση timer αποστολής δεδομένων  
    CollectDataTimer.Enabled:=false;  
    Memo1.Lines.Clear;
```





```
sl:=tstringlist.Create;  
Memo1.Lines.Add(dtstr);  
sl.Add(dtstr);  
try  
  //Μετακίνηση στο πρώτο στίγμα  
  GPSDataTable.First;  
  while (not GPSDataTable.Eof) and (GPSDataTable.RecNo<50) do begin  
    //Σύνθεση δεδομένων στίγματος  
    if GPSDataTable.Bof then  
      dtstr:='{ "name":"' +edit4.Text+', '+  
              '"lat": '+GPSDataTablelat.AsWideString+', '+  
              '"lng": '+GPSDataTablelng.AsWideString+', '+  
              '"trntime": '+FormatDateTime('yyyy-mm-dd  
hh:nn:ss.zzz',GPSDataTabletrntime.AsDateTime)+'"' +  
              '}'  
    else  
      dtstr:='{ "name":"' +edit4.Text+', '+  
              '"lat": '+GPSDataTablelat.AsWideString+', '+  
              '"lng": '+GPSDataTablelng.AsWideString+', '+  
              '"trntime": '+FormatDateTime('yyyy-mm-dd  
hh:nn:ss.zzz',GPSDataTabletrntime.AsDateTime)+'"' +  
              '}'  
    Memo1.Lines.Add(dtstr);  
    sl.Add(dtstr);  
    GPSDataTable.Edit;  
    //Ενημέρωση flag απεσταλμένου  
    GPSDataTableissent.AsInteger:=1;  
    GPSDataTable.Next;  
  
    end;  
    dtstr:=']]}';  
    Memo1.Lines.Add(dtstr);  
    sl.Add(dtstr);  
  except  
  end;  
  //Ενεργοποίηση timer αποστολής δεδομένων  
  CollectDataTimer.Enabled:=true;  
  //Δημιουργία stream δεδομένων για αποστολή, είναι το body του request  
  ssRequest:=TStringStream.Create(sl.text);  
  //Δημιουργία stream δεδομένων για την απάντηση της αποστολής  
  ssResponse:=TStringStream.Create;  
  try  
    //Post κλήση  
    IdHTTP.Post( edit1.Text+'/data',ssRequest,ssResponse);  
  
    //Φιλτράρισμα των απεσταλμένων σιγμάτων  
    GPSDataTable.Filtered:=false;  
    GPSDataTable.Filter:='issent=1';  
    GPSDataTable.Filtered:=true;  
    //Διαγραφή των απεσταλμένων σιγμάτων  
    while GPSDataTable.RecordCount>0 do  
      GPSDataTable.Delete;  
    //Ακύρωση φίλτρου απεσταλμένων  
    GPSDataTable.Filtered:=false;  
    GPSDataTable.First;  
  except  
    on e:exception do begin  
      //Μήνυμα σε περίπτωση αποτυχίας  
      Label10.Text:=e.Message;  
      ssRequest.Free;  
      ssResponse.Free;  
      exit;  
    end;  
  end;  
end;
```



```
//Parsing JSON, του response  
v:=JSONVariant(ssResponse.DataString);  
msg:=v.msg;  
//Ενημέρωση για επιτυχία ή αποτυχία  
if msg='Ok' then  
    label10.Text:='Ενημερώθηκαν '+inttostr(v.rows)+' εγγραφές...'  
else  
    label10.Text:='Αποτυχία εισαγωγής'+msg;  
//Ελευθέρωση μνήμης  
ssRequest.Free;  
ssResponse.Free;  
  
//Αποσύνδεση, δημιουργία κλήσης  
ssRequest:=TStringStream.Create('{"sessiontoken": "'+sessiontoken+'"}');  
ssResponse:=TStringStream.Create;  
try  
    IdHTTP.Post( edit1.Text+'/logout',ssRequest,ssResponse);  
    //Ελευθέρωση μνήμης  
    ssRequest.Free;  
    ssResponse.Free;  
except  
    on e:exception do begin  
        //Μήνυμα σε περίπτωση αποτυχίας  
        Label10.Text:=e.Message;  
        ssRequest.Free;  
        ssResponse.Free;  
        exit;  
    end;  
end;  
end;
```

## 4.5. Nginx Reverse Proxy

Για να υποδεχθούμε τις κλήσεις των clients στο ίδιο domain name, χρησιμοποιούμε το Nginx ως Reverse Proxy. Το “Maps” εξυπηρετεί στο Port 8077 και το “RestService” στο 8078. Μια επιπλέον εργασία που αναθέτουμε στο Nginx είναι η ασφάλεια της επικοινωνίας, όπου χρησιμοποιούμε πρωτόκολλο HTTPS. Για τον λόγο αυτό έχουμε προμηθευτεί με τα ανάλογα κλειδιά και έχουμε παραμετροποιήσει το Nginx, μέσω του παρακάτω αρχείου παραμέτρων:

```
#Πλήθος διεργασιών  
worker_processes 1;  
  
#Πλήθος συνδέσεων  
events {  
    worker_connections 1024;  
}  
  
#Παράμετροι HTTP  
http {  
    #Ορισμός Web Server, ιστοσελίδας  
    upstream webserv {  
        server 127.0.0.1:8077 max_fails=1 fail_timeout=120s;  
    }  
    #Ορισμός Rest Server  
    upstream restsrv {  
        server 127.0.0.1:8078 max_fails=1 fail_timeout=120s;  
    }  
  
    include mime.types;
```



```
default_type application/octet-stream;

sendfile on;
keepalive_timeout 65;

#Αν κάποιος χρησιμοποιήσει το HTTP, τον κατευθύνουμε στο HTTPS
server {
    listen 80;
    server_name localhost;
    rewrite ^ https://pmsapps.uth.gr$request_uri? permanent;
    error_page 500 502 503 504 /50x.html;
}

# HTTPS server
server {

    listen 443 ssl;
    server_name localhost;

    gzip on;
    gzip_proxied any;
    gzip_types
    text/css
    text/*
    text/javascript
    message/*
    application/x-javascript
    application/json
    application/xml
    application/atom+xml
    application/xaml+xml;

    root c:/pms/nginx-1.11.11/;

    #Παράμετροι για κενά ασφαλείας
    ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
    ssl_ciphers "EECDH+AESGCM:EDH+AESGCM:ECDHE-RSA-AES128-GCM-SHA256:AES256+EECDH:DHE-RSA-AES128-GCM-SHA256:AES256+EDH:ECDHE-RSA-AES256-GCM-SHA384:DHE-RSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-SHA384:ECDHE-RSA-AES128-SHA256:ECDHE-RSA-AES256-SHA:ECDHE-RSA-AES128-SHA:DHE-RSA-AES256-SHA256:DHE-RSA-AES128-SHA256:DHE-RSA-AES256-SHA:DHE-RSA-AES128-SHA:ECDHE-RSA-DES-CBC3-SHA:EDH-RSA-DES-CBC3-SHA:AES256-GCM-SHA384:AES128-GCM-SHA256:AES256-SHA256:AES128-SHA256:AES256-SHA:DES-CBC3-SHA:HIGH:!aNULL:!eNULL:!EXPORT:!DES:!MD5:!PSK:!RC4";
    ssl_session_cache shared:SSL:10m;
    ssl_session_timeout 10m;
    ssl_prefer_server_ciphers on;

    #Αρχεία καταγραφής
    access_log c:/pms/nginx-1.11.11/logs/https.access.log;
    error_log c:/pms/nginx-1.11.11/logs/https.error.log;

    #Προεπιλεγμένο character set
    charset utf-8;

    #Κλειδιά SSL
    ssl_certificate c:/pms/nginx-1.11.11/ssl/bundle.crt;
    ssl_certificate_key c:/pms/nginx-1.11.11/ssl/pmsapps_uth_gr.key;

    error_page 404 /oops.htm;
    error_page 500 502 503 504 /oops.html;
```



```
#Αν το URL δείχνει το path /rest/, τότε εξυπηρετεί το RestService
location /rest/ {
    proxy_pass http://restsrv;
    proxy_redirect http://restsrv/ /;
    proxy_read_timeout      600;
    send_timeout            600;
    proxy_set_header        Host            $host;
    proxy_set_header        X-Real-IP       $remote_addr;
    proxy_set_header        X-Forwarded-For $proxy_add_x_forwarded_for;
}

#Διαφορετικά εξυπηρετεί το Maps
location / {
    proxy_pass http://websrv;
    proxy_redirect http://websrv/ /;
    proxy_read_timeout      600;
    send_timeout            600;
    proxy_set_header        Host            $host;
    proxy_set_header        X-Real-IP       $remote_addr;
    proxy_set_header        X-Forwarded-For $proxy_add_x_forwarded_for;
}
}
```



*Βελέντζας Πολυχρόνης*  
*Διαχείριση Γεωγραφικών Ερωτημάτων και*  
*Υπηρεσία Διαδραστικών Χαρτών για Ιστοσελίδες*

## 5. Παρουσίαση Χρήσης Πλατφόρμας

Στο κεφάλαιο αυτό θα παρουσιάσουμε την χρήση της Πλατφόρμας και τις εφαρμογές που την αποτελούν. Όπως περιγράψαμε και στο κεφάλαιο 1, σκοπός της διατριβής είναι η δημιουργία ενός γεωγραφικού συστήματος, όπου θα μπορεί ο διαχειριστής να σχεδιάσει ερωτήματα και να τα παρουσιάσει, μέσω του διαδικτύου, σε φορητές και μη συσκευές. Παρακάτω θα περιγράψουμε την λειτουργία της ιστοσελίδας, τον σχεδιασμό γεωγραφικών ερωτημάτων, την διαδικτυακή εφαρμογή προβολής των σελίδων, την υπηρεσία REST και την βοηθητική εφαρμογή κινητών καταγραφής στίγματος.

### 5.1. Ιστοσελίδα

Η ιστοσελίδα [34] βοηθά τον διαχειριστή να ανοίξει ένα νέο λογαριασμό, να εισέλθει στο σύστημα και τέλος να διαχειριστεί τα γεωγραφικά του ερωτήματα. Η αρχική σελίδα που συναντά ο χρήστης τον πληροφορεί ότι πρόκειται για την παρούσα μεταπτυχιακή διατριβή. Η σελίδα έχει την παρακάτω μορφή:



Εικόνα 5.1.1 Κεντρική σελίδα ιστοσελίδας

Πάνω δεξιά παρουσιάζεται το μενού της σελίδας με τις επιλογές:

- ❖ Sign Up
- ❖ Login
- ❖ Πληροφορίες





#### ❖ Αρχική Σελίδα

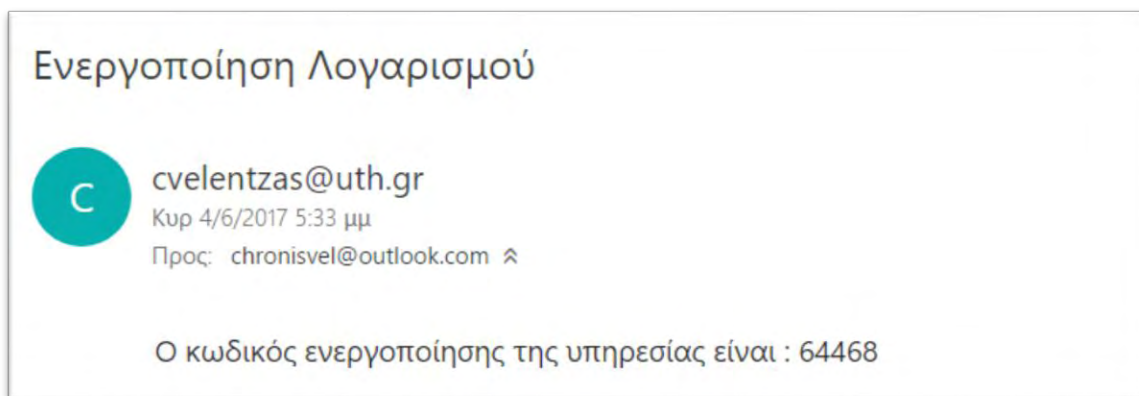
Πατώντας ο χρήστης την επιλογή “Sign Up” τον κατευθύνει ή εφαρμογή σε μια φόρμα καταχώρησης όπου απαιτεί το E-mail, το ονοματεπώνυμο τον επιθυμητό κωδικό εισόδου, την επιβεβαίωση του κωδικού εισόδου και τέλος αν επιθυμεί ο χρήστης την δημιουργία δεδομένων επίδειξης.

Εικόνα 5.1.2 Sign Up

Πατώντας «Δημιουργία Λογαριασμού» το σύστημα αποστέλλει ένα e-mail ενεργοποίησης στο λογαριασμό που συμπληρώθηκε από την χρήστη. Στην συνέχεια εμφανίζεται η σελίδα που ζητά τον κωδικό ενεργοποίησης που έστειλε με e-mail στον χρήστη.

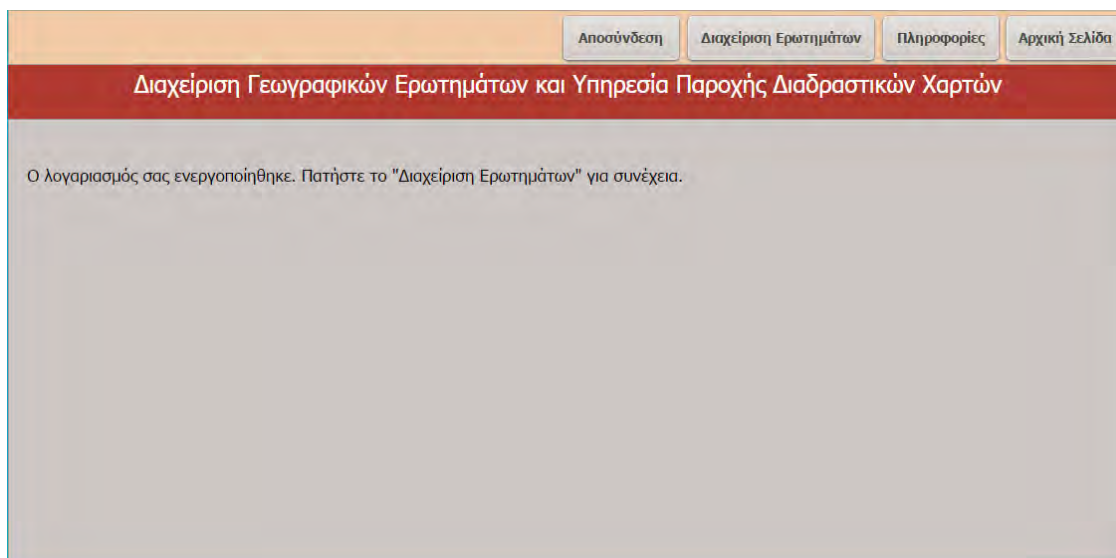
Εικόνα 5.1.3 Ενεργοποίηση Λογαριασμού

Το e-mail που λαμβάνει ο χρήστης έχει αποστολέα τον [cvelentzas@uth.gr](mailto:cvelentzas@uth.gr), η οποία είναι η ηλεκτρονική διεύθυνση του Χρόνη Βελέντζα και έχει την μορφή:



Εικόνα 5.1.4 Email Ενεργοποίησης

Εισάγοντας ο χρήστης τον κωδικό στη παραπάνω φόρμα εισέρχεται στο σύστημα. Μετά την επιτυχημένη ενεργοποίηση εμφανίζεται το κουμπί «**Διαχείριση Ερωτημάτων**»



Εικόνα 5.1.5 Επιτυχημένη ενεργοποίηση λογαριασμού.

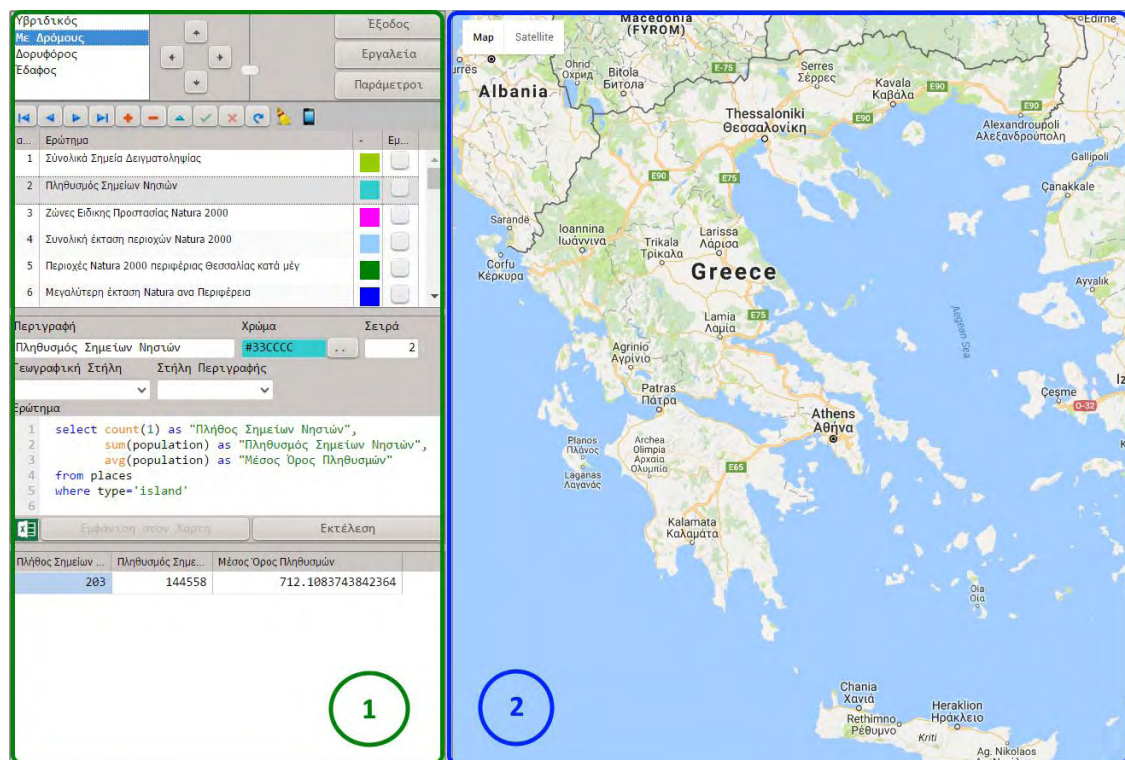
Πατώντας το κουμπί «**Αποσύνδεση**» ο χρήστης αποσυνδέεται και εμφανίζεται ξανά η αρχική σελίδα. Για να κάνει πάλι εισαγωγή στο σύστημα θα πρέπει να πατήσει το “**Login**”, το οποίο τον προτρέπει να εισάγει το e-mail και τον κωδικό εισόδου του.

Εικόνα 5.1.6 Login

Μετά την είσοδο στην σελίδα μπορεί ο χρήστης να προχωρήσει στην σχεδίαση γεωγραφικών ερωτημάτων πατώντας το «Διαχείριση Ερωτημάτων».

## 5.2. Διαχείριση Ερωτημάτων

Μετά την είσοδο του χρήστη εμφανίζεται αριστερά ένα πάνελ και δεξιά ο χάρτης Google maps. Στο πάνελ αριστερά ο χρήστης μπορεί να δημιουργήσει και να δοκιμάσει ερωτήματα, γεωγραφικά και μη. Στον χάρτη μπορεί παράλληλα να προβάλει την γραφική αναπαράστασή τους.



**Ερωτήματα**

Ερώτημα	Εμ...
1. Συνολικά Σημεία Δεγματοληψίας	
2. Πληθυσμός Σημείων Νησιών	
3. Ζώνες Ειδικής Προστασίας Natura 2000	
4. Συνολική έκταση περιοχών Natura 2000	
5. Περιοχές Natura 2000 περιφέρειας Θεσσαλίας κατά μέγ	
6. Μεγαλύτερη έκταση Natura ανα Περιφέρεια	

**Περιγραφή**

Χρώμα: #33CCCC Σελίδα: 2

Πληθυσμός Σημείων Νησιών

Γεωγραφική Στήλη Στήλη Περιγραφής

**Ερώτημα**

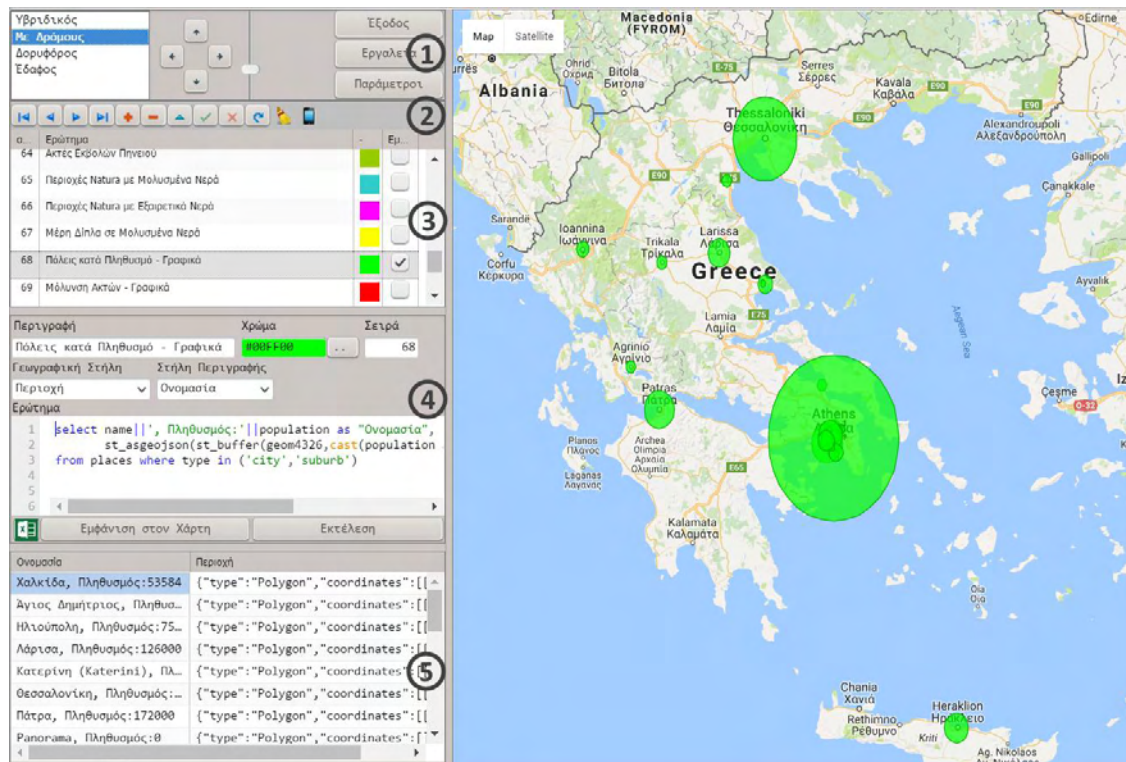
```
1. select count(1) as "Πλήθος Σημείων Νησιών",
2. sum(population) as "Πληθυσμός Σημείων Νησιών",
3. avg(population) as "Μέσος Όρος Πληθυσμών"
4. from places
5. where type='island'
```

**Εκτέλεση**

Πληθυσμός Σημείων...	Πληθυσμός Σημ...	Μέσος Όρος Πληθυσμών
203	144558	712.1083743842364

Εικόνα 5.2.1 Προβολή Διαχείρισης Ερωτημάτων (1:Πάνελ διαχείρισης, 2:Χάρτης)

Στην συνέχεια θα περιγράψουμε αναλυτικά το πάνελ διαχείρισης καθώς και τα εργαλεία και τις παραμέτρους του.



Εικόνα 5.2.2 Αναλυτικά μέρη του πάνελ διαχείρισης

## Κυρίως Πάνελ

Το πάνελ χωρίζεται στα παρακάτω διαχειριστικά μέρη :

### 1. Χειριστήρια χάρτη

Τα χειριστήρια του χάρτη παρέχουν μια διεπαφή χρήστη για την βασική προβολή και πλοήγηση του χάρτη. Οι διαθέσιμες προβολές είναι οι:

- ❖ Με Δρόμους  
Είναι η προεπιλεγμένη προβολή χαρτών και η μορφή της μοιάζει με τους οδικούς χάρτες
- ❖ Δορυφόρος  
Παρουσιάζει φωτογραφία δορυφόρου στην προβαλλόμενη περιοχή
- ❖ Έδαφος  
Είναι μια μορφή ανάγλυφου που μας δίνει την αίσθηση ύψους των λόφων και βουνών της περιοχής
- ❖ Υβριδικός  
Είναι ο χάρτης δορυφόρου μαζί με τον οδικό χάρτη


















Ο χρήστης μπορεί να επιλέξει μια από τις διαθέσιμες και αυτομάτως ο χάρτης ενημερώνεται με την νέα εικόνα.

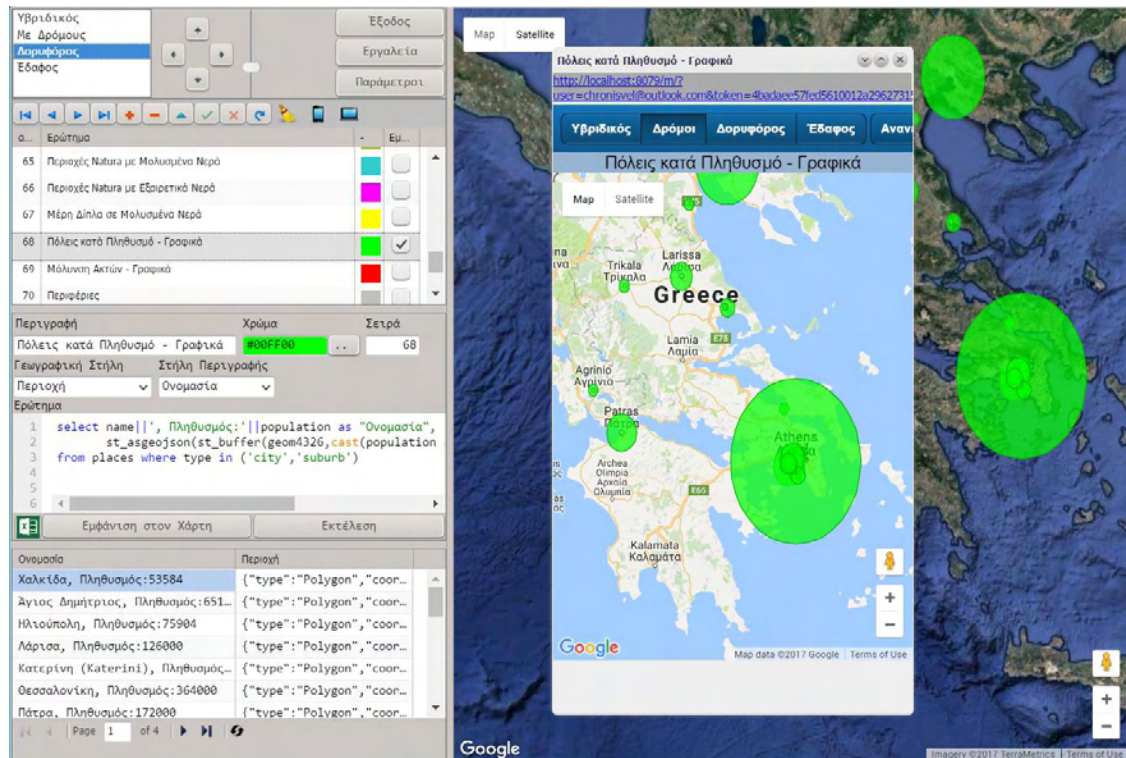
Δεξιότερα εμφανίζονται τα πλήκτρα μετακίνησης του χάρτη, που επιτρέπουν την κύλιση του χάρτη πάνω, κάτω και δεξιά αριστερά. Επιπλέον υπάρχει η γραμμή κύλισης που καθορίζει την επιθυμητή κλίμακα του χάρτη.

Το «Έξοδος» μας επιστρέφει στην αρχική σελίδα της ιστοσελίδας. Το «Εργαλεία» και «Παράμετροι» θα αναλυθούν παρακάτω.

## 2. Εργαλειοθήκη ερωτημάτων

Η εργαλειοθήκη εκτελεί διαχειριστικές λειτουργίες στην λίστα ερωτημάτων. Παρέχει τις εξής λειτουργίες:

-  Μετακίνηση στην πρώτη εγγραφή
-  Μετακίνηση μια εγγραφή πίσω
-  Μετακίνηση μια εγγραφή μπροστά
-  Μετακίνηση στην τελευταία εγγραφή
-  Εισαγωγή νέας εγγραφής
-  Διαγραφή εγγραφής (προηγείται ερώτηση επιβεβαίωσης)
-  Επεξεργασία εγγραφής (εκτελείται αυτόματα)
-  Καταχώρηση εγγραφής (εκτελείται αυτόματα)
-  Ακύρωση μεταβολής εγγραφής (εφόσον δεν έχει καταχωρηθεί)
-  Ανανέωση λίστας
-  Καθαρισμός Χάρτη, καθαρίζει τα προβαλλόμενα γεωγραφικά ερωτήματα
-  Προβολή Χάρτη σε μέγεθος κινητού (μέσα σε νέο παράθυρο)
-  Προβολή Χάρτη σε νέο παράθυρο



Εικόνα 5.2.3 Προεπισκόπηση Χάρτη σε μορφή κινητού

### 3. Λίστα ερωτημάτων

Τα διαθέσιμα ερωτήματα εμφανίζονται σε αυτό το τμήμα. Ο χρήστης μπορεί να επιλέξει ένα ερώτημα και να το προβάλει στον χάρτη με διπλό κλικ ή τσεκάροντας την αντίστοιχη στήλη «Εμφάνιση». Επιπλέον εμφανίζεται το προεπιλεγμένο χρώμα εμφάνισης του ερωτήματος.

### 4. Επεξεργασία ερωτήματος

Στην επεξεργασία ερωτήματος ο χρήστης σχεδιάζει το γεωγραφικό του ερώτημα. Αναλυτικότερα σε αυτό το σημείο συμπληρώνουμε τα παρακάτω στοιχεία:

- ❖ Περιγραφή  
Είναι η περιγραφή του ερωτήματος, η οποία εμφανίζεται στις προβολές των συσκευών
- ❖ Χρώμα  
Επιλέγουμε το επιθυμητό χρώμα με το οποίο θα προβληθεί το γεωγραφικό μας ερώτημα
- ❖ Σειρά  
Καθορίζουμε την σειρά εμφάνισης του ερωτήματος
- ❖ Γεωγραφική Στήλη  
Εφόσον το ερώτημά μας περιέχει γεωγραφικά δεδομένα, σε αυτή την στήλη εμφανίζονται τα ονόματα των πεδίων με γεωγραφικό περιεχόμενο
- ❖ Στήλη περιγραφής

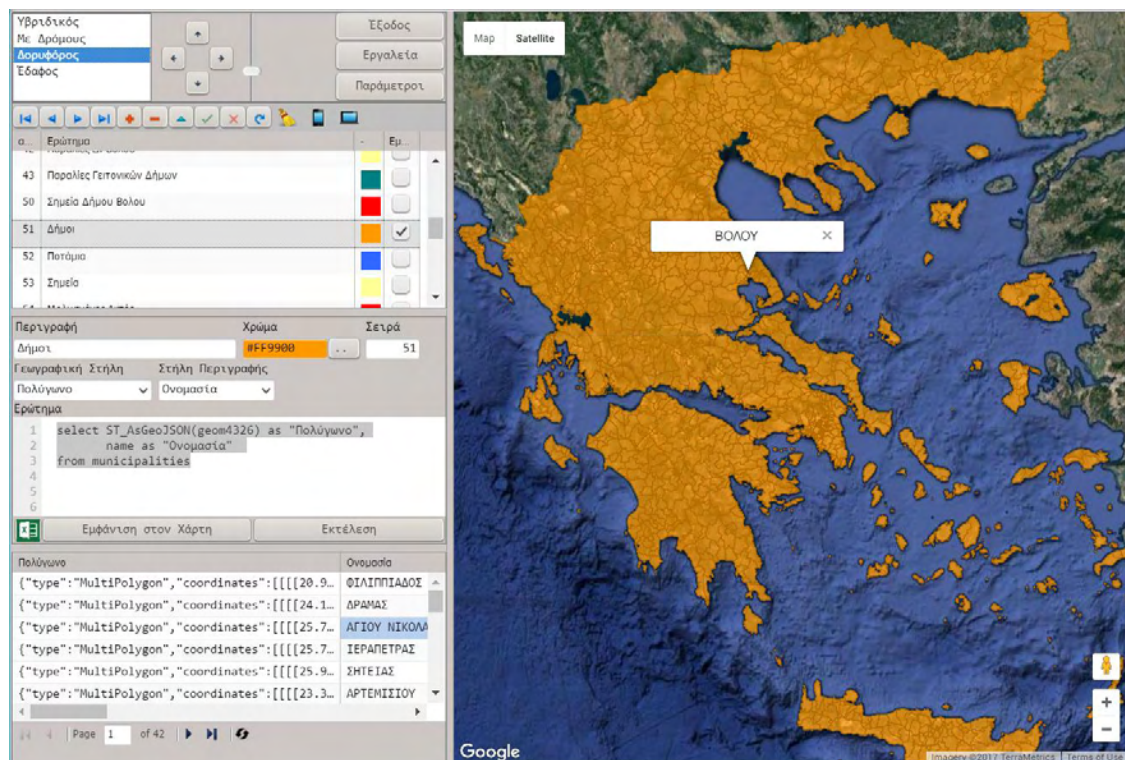


Εφόσον το ερώτημά μας περιέχει δεδομένα κειμένου, αυτή η λίστα εμφανίζει τα αντίστοιχα πεδία

❖ Ερώτημα

Στο πεδίο «Ερώτημα» συντάσσουμε το ερώτημα σε μορφή SQL. Η σύνταξη των ερωτημάτων ακολουθούν τα πρότυπα της PostgreSQL 9.5 με την επέκταση GIS (PostGIS). Το ερώτημα θα πρέπει να επιστρέφει γεωγραφικά δεδομένα εφαρμόζοντας στην γεωγραφική στήλη την συνάρτηση ST\_AsGeoJSON. Η συνάρτηση αυτή επιστρέφει κατάλληλο περιεχόμενο για προβολή των δεδομένων στον χάρτη της Google. Για παράδειγμα το παρακάτω ερώτημα που εμφανίζει τους Δήμους της Ελλάδας :

```
select ST_AsGeoJSON(geom4326) as "Πολύγωνο",  
       name as "Ονομασία"  
from municipalities
```



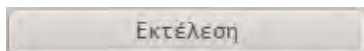
Εικόνα 5.2.4 Εμφάνιση ερωτήματος «Δήμοι»

Εδώ βλέπουμε ότι η στήλη «Πολύγωνο» αναγνωρίστηκε ως γεωγραφική στήλη και η στήλη «Ονομασία» ως στήλη περιγραφής. Η εφαρμογή χρησιμοποιεί την γεωγραφική στήλη για να σχεδιάσει τα δεδομένα πάνω στον χάρτη και την στήλη περιγραφής για να μας πληροφορήσει τί είναι αυτό που βλέπουμε, κάνοντας κλικ πάνω του. Στην παραπάνω εικόνα πατήσαμε στον Δήμο Βόλου.

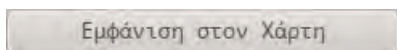
Επιπλέον εμφανίζονται τα παρακάτω πλήκτρα :



Δημιουργεί ένα Αρχείο Excel με το αποτέλεσμα του ερωτήματος, το οποίο κατεβάζει ο χρήστης



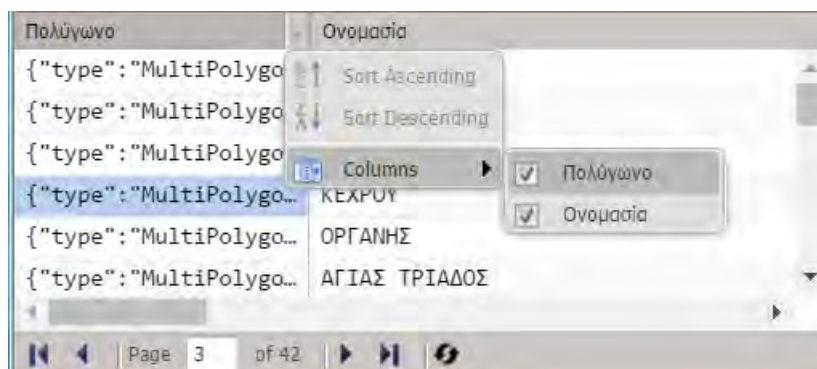
Εκτελεί το ερώτημα του χρήστη και εμφανίζει το αποτέλεσμα στη θέση «5. Δεδομένα ερωτήματος»



Εμφανίζει το ερώτημα στον χάρτη






### 5. Δεδομένα ερωτήματος

Κάθε φορά που εκτελείται ένα ερώτημα, το αποτέλεσμα του εμφανίζεται σε αυτή την θέση. Με αυτό τον τρόπο ο χρήστης έχει την δυνατότητα να πληροφορείται για την ορθότητα και την μορφή του ερωτήματός του. Επειδή τα επιστρεφόμενα δεδομένα είναι συνήθως πολλά, το αποτέλεσμά τους πρέπει να εμφανισθεί σελιδοποιημένο. Η σελιδοποίηση στο προβαλλόμενο grid έχει μέγεθος 25 γραμμών.



Εικόνα 5.2.5 Grid Δεδομένων Ερωτήματος

Στο grid μπορούμε να ταξινομήσουμε τα δεδομένα μας καθώς και να προσθαφαιρέσουμε στήλες. Επιπλέον το grid έχει τα παρακάτω στοιχεία ελέγχου :

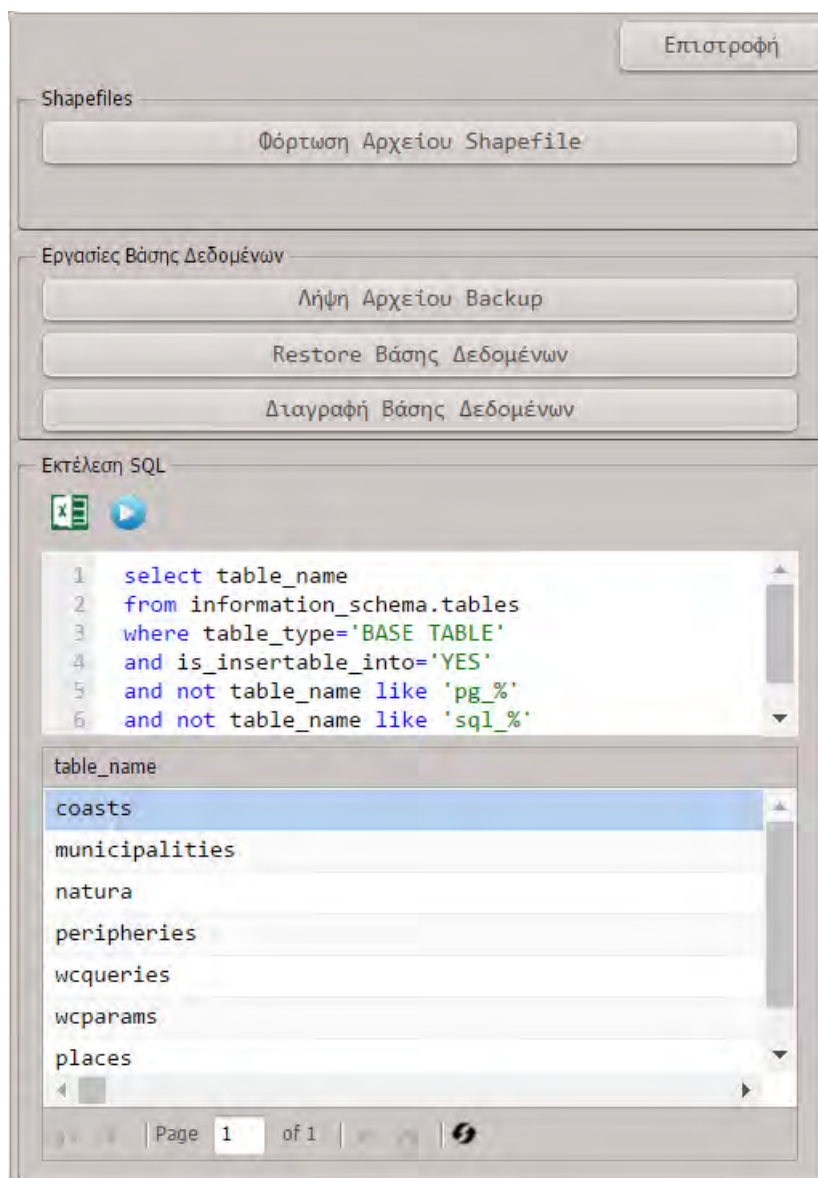
-  Μετακίνηση στην πρώτη εγγραφή
-  Μετακίνηση στην προηγούμενη εγγραφή
-  Μετακίνηση στην επόμενη εγγραφή
-  Μετακίνηση στην τελευταία εγγραφή
-  Ανανέωση δεδομένων

Page 3 of 42 Προβολή τρέχουσας σελίδας

Εκτός από τον σχεδιασμό και την διαχείριση γεωγραφικών ερωτημάτων στο κυρίως πάνελ υπάρχουν ακόμη δύο πάνελ τα «Εργαλεία» και «Παράμετροι»

### Εργαλεία

Πατώντας το πλήκτρο «Εργαλεία» εμφανίζεται το πάνελ εργαλείων της εφαρμογής. Εδώ ο χρήστης μπορεί να εκτελέσει μερικές διαχειριστικές λειτουργίες καθώς επίσης να δοκιμάσει εντολές SQL.



Εικόνα 5.2.6 Πάνελ Εργαλείων

Το πάνελ εργαλείων χωρίζεται στις εξής περιοχές:



❖ Shapefiles [35]

Τα αρχεία τύπου shapefile περιέχουν δεδομένα γεωγραφικών διανυσμάτων. Είναι το δημοφιλέστερο format ανταλλαγής δεδομένων. Αναπτύχθηκε και τυποποιήθηκε από το Esri με στόχο την διαλειτουργικότητα και την ανταλλαγή δεδομένων μεταξύ λοιπών GIS συστημάτων.

Με την επιλογή αυτή ο χρήστης μπορεί να ανεβάσει ένα αρχείο τύπου shapefile και να το εισάγει στην βάση δεδομένων.

❖ Εργασίες Βάσης Δεδομένων

Στην θέση αυτή η εφαρμογή υλοποιεί τις παρακάτω τρεις βασικές λειτουργίες:

1. Λήψη Αρχείου Backup

Εκτελώντας αυτή την λειτουργία, ο server δημιουργεί ένα αρχείο backup και το επιστρέφει στον χρήστη. Το αρχείο δημιουργείται με την εκτέλεση της παρακάτω εντολής :

```
pg_dump.exe --host 127.0.0.1 --port 5432 --username "postgres" --format=c --no-acl --no-owner --no-password --verbose --file ".\temp\pg.backup" --schema "\"user_XXXXXXXXXX\" \"postgres"
```

2. Restore Βάσης Δεδομένων

Ο χρήστης μπορεί να ανεβάσει στον server ένα αρχείο backup και να το κάνει restore

3. Διαγραφή Βάσης Δεδομένων

Η επιλογή αυτή «καθαρίζει» την βάση δεδομένων του χρήστη και διαγράφει όλους τους πίνακές του.

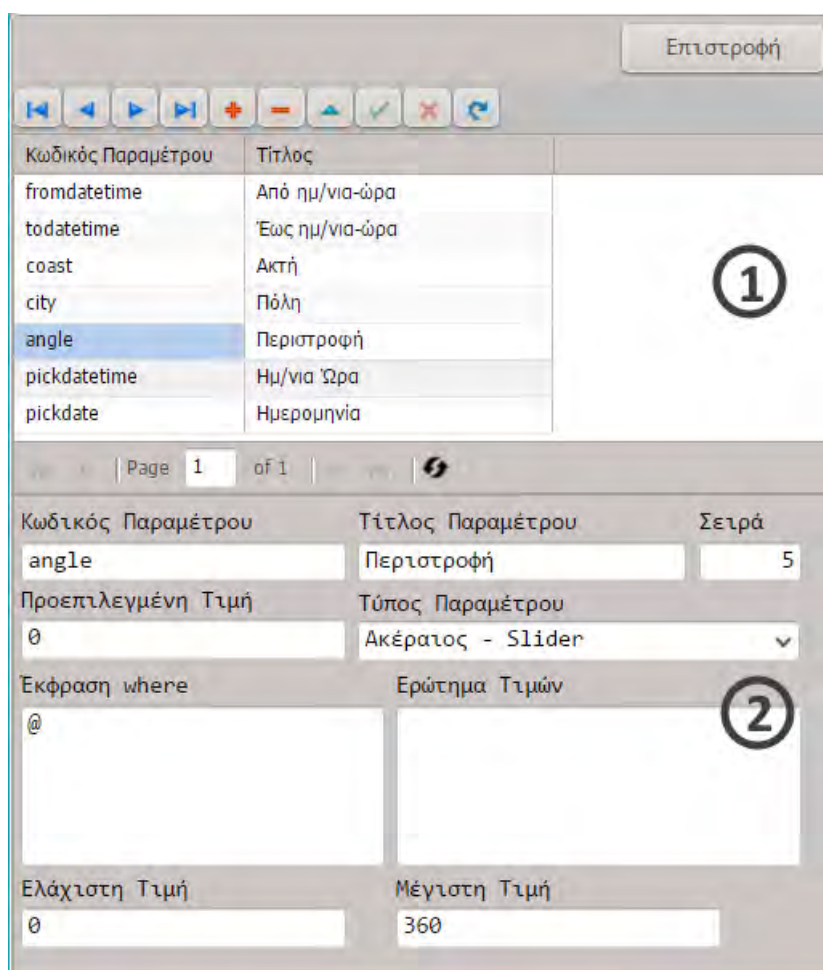
❖ Εκτέλεση SQL

Εδώ ο χρήστης έχει την ελευθερία να εκτελέσει οποιαδήποτε εντολή SQL και να πληροφορηθεί για το αποτέλεσμα της. Ενδεικτικά θα μπορούσε να τρέξει την παρακάτω εντολή, η οποία επιστρέφει όλους τους πίνακές του:

```
select table_name  
from information_schema.tables  
where table_type='BASE TABLE'  
and is_insertable_into='YES'  
and not table_name like 'pg_%'  
and not table_name like 'sql_%'  
and not table_name like 'spatial_%'
```

### Παράμετροι

Το πάνελ «Παράμετροι» παρέχει στον χρήστη την δυνατότητα να δημιουργήσει «φίλτρα» επιλογών και να τα συνδυάσει με τα γεωγραφικά ερωτήματα. Η εφαρμογή με την δυνατότητα των παραμέτρων δημιουργεί ένα αλληλεπιδραστικό περιβάλλον με τον τελικό χρήστη της εφαρμογής, δηλαδή τον χρήστη που θα χρησιμοποιήσει την «Υπηρεσία Διαδραστικών Χαρτών».



Επιστροφή

Κωδικός Παραμέτρου    Τίτλος

fromdatetime	Από ημ/νια-ώρα
todatetime	Έως ημ/νια-ώρα
coast	Ακτή
city	Πόλη
angle	Περιστροφή
pickdatetime	Ημ/νια Ώρα
pickdate	Ημερομηνία

Page 1 of 1

Κωδικός Παραμέτρου    Τίτλος Παραμέτρου    Σειρά

angle	Περιστροφή	5
-------	------------	---

Προεπιλεγμένη Τιμή    Τύπος Παραμέτρου

0    Ακέραιος - Slider

Εκφραση where    Ερώτημα Τιμών

@

Ελάχιστη Τιμή    Μέγιστη Τιμή

0    360

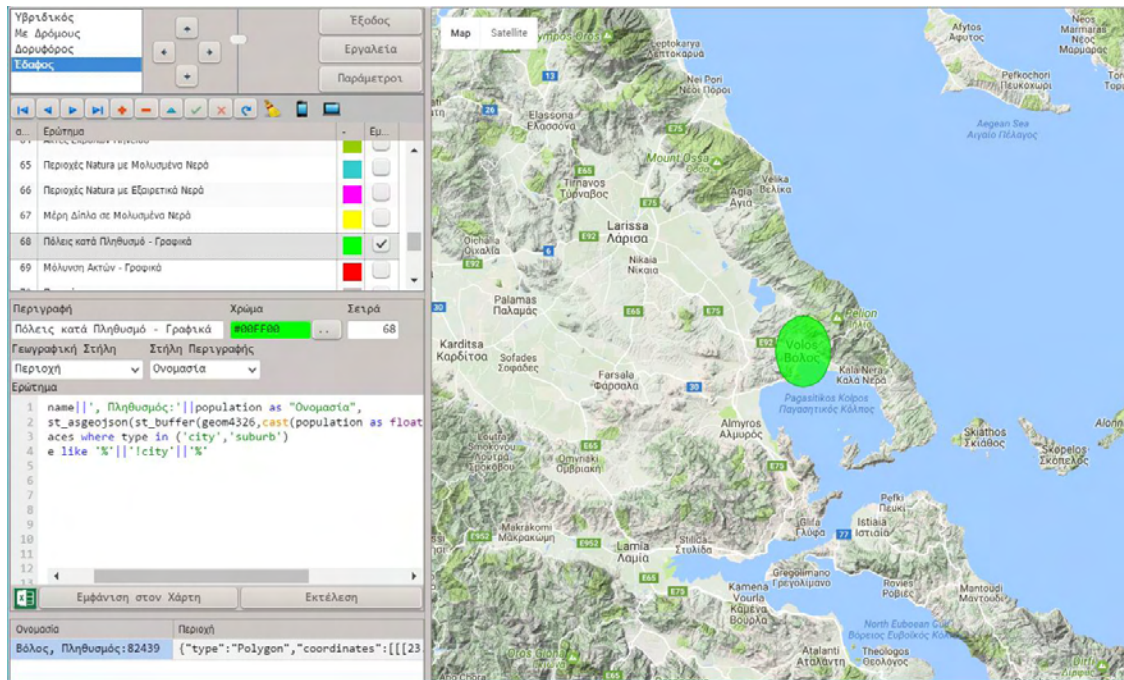
Εικόνα 5.2.7 Παράμετροι, θέσεις (1), (2)

Πατώντας το «Παράμετροι» εμφανίζεται στην θέση (1) μια εργαλειοθήκη και μια λίστα με τις παραμέτρους του χρήστη. Στην θέση (2) μπορούμε να επεξεργαστούμε μια παράμετρο. Η φόρμα αυτή περιέχει τα εξής πεδία:

❖ Κωδικός Παραμέτρου

Είναι ο κωδικός του φίλτρου που εισάγουμε στο ερώτημα. Η εφαρμογή εκτελώντας ένα ερώτημα αναζητά μεταβλητές τύπου macros και αν υπάρχουν ενσωματώνει την παράμετρο σε αυτές. Ας δούμε το παρακάτω παράδειγμα :





Εικόνα 5.2.8 Παράδειγμα παραμέτρου “city”

Το ερώτημα που έχουμε σχεδιάσει είναι το :

```
select name||', Πληθυσμός: '||population as "Όνομασία",
st_asgeojson(st_buffer(geom4326,cast(population as float)/1000000)) as "Π
εριοχή"
from places where type in ('city','suburb')
and name like '%||!city||%'
```

Σε αυτό παρατηρούμε ότι περιέχει το κείμενο “!city”. Αυτό αποτελεί μια macro και πριν την εκτέλεση του ερωτήματος θα πρέπει να αντικατασταθεί με κάποιο συστατικό μέρος, από τους παραμέτρους, για να μπορέσει να εκτελεσθεί χωρίς σφάλματα.

Στο προηγούμενο παράδειγμα, η εφαρμογή αναζήτησε στις παραμέτρους μια παράμετρο με κωδικό “city” και τροφοδότησε το ερώτημα με την προεπιλεγμένη της τιμή που είναι σε αυτή την περίπτωση το «Βόλος».

Όταν αυτό το ερώτημα εκτελεσθεί στην «Υπηρεσία Διαδραστικών Χαρτών» όμοια θα προσπαθήσει να αντικαταστήσει τα macros με τις προεπιλεγμένες τιμές ή με τιμές που θα εισάγει ο τελικός χρήστης, διαμέσου δυναμικών στοιχείων ελέγχου που θα δημιουργηθούν για αυτό τον λόγο. Περισσότερα θα περιγράψουμε στην ενότητα «Υπηρεσία Διαδραστικών Χαρτών» 5.3.

#### ❖ Τίτλος Παραμέτρου

Στο πεδίο αυτό εισάγουμε τον τίτλο της παραμέτρου, ο οποίος θα εμφανίζεται στα δυναμικά στοιχεία ελέγχου της «Υπηρεσίας Διαδραστικών Χαρτών»

#### ❖ Σειρά

Εισάγουμε την σειρά που θα εμφανιστεί το συγκεκριμένο στοιχείο ελέγχου





❖ Προεπιλεγμένη Τιμή

Είναι η προεπιλεγμένη τιμή της παραμέτρου και χρησιμοποιείται όταν εκτελούμε τα γεωγραφικά ερωτήματα, στο πάνελ διαχείρισης.

❖ Τύπος Παραμέτρου

Υπάρχουν πολλοί τύποι παραμέτρων, ανάλογα με το είδους των δεδομένων (κείμενο, ακέραιος, ημερομηνία κτλ), καθώς επίσης και με το είδος εισαγωγής τιμών από το δυναμικό στοιχείο ελέγχου (Πεδίο κειμένου, Ημερολόγιο, Checkbox, Λίστα τιμών κτλ). Οι τύποι που υποστηρίζονται, δημιουργούν δυναμικά στοιχεία ελέγχου κατά τον χρόνο εκτέλεσης και είναι οι παρακάτω :

1. Κείμενο

Πεδίο ελεύθερου κειμένου

2. Ακέραιος

Πεδίο που δέχεται ως είσοδο μόνο ακέραιους αριθμούς

3. Πραγματικός

Πεδίο που δέχεται ως είσοδο μόνο πραγματικούς αριθμούς

4. Ημερομηνία

Πεδίο που δέχεται ως είσοδο ημερομηνία

5. Ναι/Όχι

Checkbox, επιστρέφει «1» για το «Ναι» και «0» για το «Όχι»

6. Λίστα Τιμών

Πτυσσόμενη λίστα, από την οποία ο χρήστης επιλέγει ένα στοιχείο

7. Λίστα Πολλαπλής Επιλογής

Λίστα από την οποία ο χρήστης μπορεί να επιλέξει πολλά στοιχεία ταυτόχρονα

8. Ημερομηνία-Ωρα

Πεδίο που δέχεται ως είσοδο ημερομηνία-ώρα

9. Ακέραιος-Slider

Γραμμή κύλισης που επιστρέφει ακέραιο αριθμό

❖ Έκφραση where

Η έκφραση where μας βοηθά να σχηματίσουμε αποτελεσματικότερα το κείμενο με το οποίο θα αντικαταστήσουμε την εκάστοτε macro. Θα μας βοηθούσε για παράδειγμα να μετατρέψουμε ένα κείμενο σε ημερομηνία, όπως το παράδειγμα παρακάτω:

Κωδικός Παραμέτρου	Τίτλος Παραμέτρου	Σειρά
fromdatetime	Από ημ/νια-ώρα	1
Προεπιλεγμένη Τιμή	Τύπος Παραμέτρου	
2017-01-01 01:00:00.000	Ημερομηνία-Ωρα	
Εκφραση where	Ερώτημα Τιμών	
cast('@' as timestamp)::timestamp without time zone		
Ελάχιστη Τιμή	Μέγιστη Τιμή	

Εικόνα 5.2.9

Σε αυτό το παράδειγμα το κείμενο μετατρέπεται σε ημερομηνία με το:

```
cast('@' as timestamp)::timestamp without time zone
```

Δεν απαιτείται βέβαια πάντα μια ανάλογη μετατροπή. Στις περιπτώσεις που θα χρησιμοποιήσουμε αυτούσια την τιμή εισαγωγής τότε στο πεδίο «Έκφραση where» εισάγουμε το «@». Η εφαρμογή δηλαδή αντικαθιστά το «@» με την προεπιλεγμένη τιμή ή την τιμή που εισάγει ο χρήστης.

- ❖ Ερώτημα τιμών  
Ενεργοποιείται στους τύπους παραμέτρων «Λίστα Τιμών» και «Λίστα Πολλαπλής Επιλογής». Στο ερώτημα τιμών θα μπορούσε κανείς να εισάγει ένα ερώτημα με το αποτέλεσμα που θα ήθελε να εμφανισθεί στην «Λίστα Τιμών»
- ❖ Ελάχιστη τιμή  
Χρησιμοποιείται στον τύπο παραμέτρων «Ακέραιος-Slider». Είναι η ελάχιστη τιμή του Slider και είναι πάντα ακέραιος.
- ❖ Μέγιστη τιμή  
Χρησιμοποιείται επίσης στον τύπο παραμέτρων «Ακέραιος-Slider». Είναι η μέγιστη τιμή του Slider και είναι πάντα ακέραιος.

### 5.3. Υπηρεσία Διαδραστικών Χαρτών

Η υπηρεσία διαδραστικών χαρτών, παρέχει στον χρήστη το τελικό αποτέλεσμα, δηλαδή τον χάρτη με την απεικόνιση του γεωγραφικού ερωτήματος. Ένας προγραμματιστής θα μπορούσε εύκολα να ενσωματώσει αυτή την υπηρεσία στην εφαρμογή του ή στην ιστοσελίδα του. Η ενσωμάτωση της υπηρεσίας θα μπορούσε να γίνει με δύο τρόπους:

1. Σε πλήρη σελίδα  
Όταν θέλουμε να εμφανίσουμε τον χάρτη σε πλήρη σελίδα τότε χρησιμοποιούμε μόνο την διεύθυνση του server ως url, πχ:

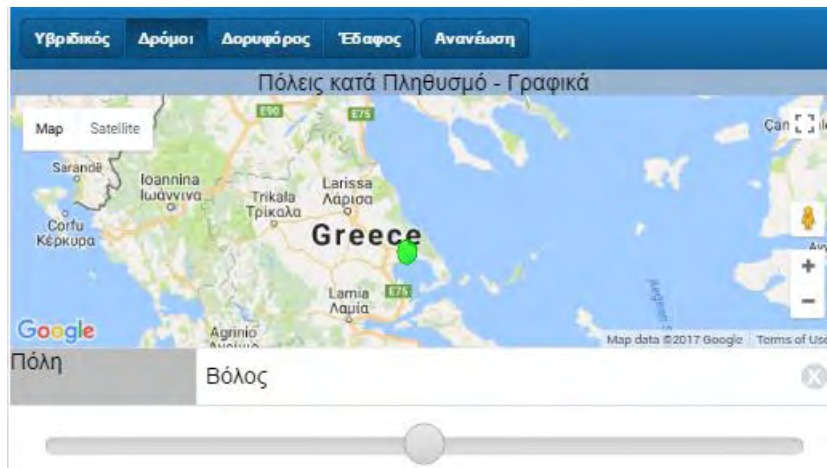
<http://servername:8079/m/?user=USER-EMAIL&token=XXXXXX&zoom=7&report=68>

## 2. Σε περιοχή σελίδας

Σε περιοχή σελίδας μπορούμε να το ενσωματώσουμε με την χρήση του tag iframe, πχ:

```
<iframe src = "http://servername: 8079/m/? user = USER-EMAIL & token = XXXXXX  
& zoom= 7 & report= 68" > </iframe>
```

Όπου servername είναι το όνομα ή το IP του server μας. Παρακάτω εμφανίζεται παράδειγμα της προβολής του της υπηρεσίας για το γεωγραφικό ερώτημα «Πόλεις κατά Πληθυσμό - Γραφικά»



Εικόνα 5.3.1 Παράδειγμα Προβολής Διαδραστικού χάρτη σε Κινητό Τηλέφωνο

Στο πάνω μέρος της προβολής εμφανίζεται εργαλειοθήκη με στοιχεία ελέγχου του χάρτη. Έχουν υλοποιηθεί οι γνωστές προβολές χάρτη Υβριδικός, Δρόμοι, Δορυφόρος και Έδαφος. Υπάρχει και ένα πλήκτρο Ανανέωση που μπορεί να ανανεώνει και να ξανασχεδιάζει τον χάρτη.

Στο κάτω μέρος της προβολής εμφανίζεται μια γραμμή κύλισης η οποία ρυθμίζει την κλίμακα του χάρτη. Ο χάρτης μπορεί να μετατοπισθεί είτε με το ποντίκι (στις περιπτώσεις επιτραπέζιων υπολογιστών) είτε με το ανάλογο gesture του δάκτυλου πάνω στην οθόνη.

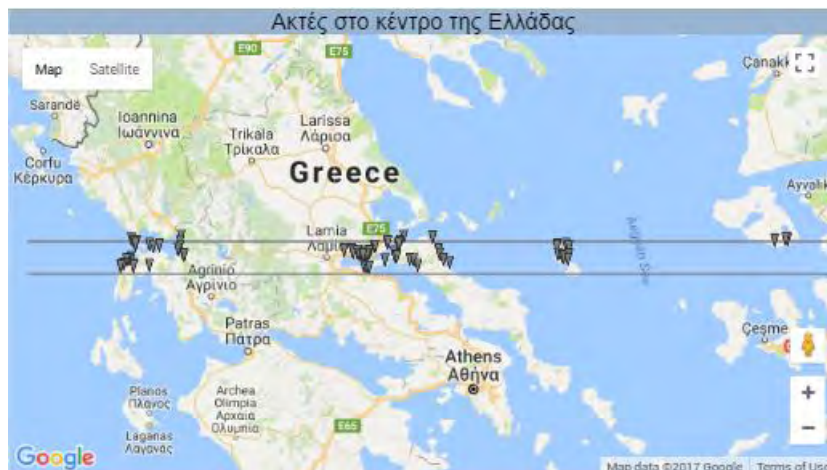
Ακριβώς πάνω από την γραμμή κύλισης η υπηρεσία διαδραστικών χαρτών δημιουργεί δυναμικά, κατά τον χρόνο εκτέλεσης της εφαρμογής, όλες τις παραμέτρους που συμμετέχουν στην εκτέλεση του ερωτήματος. Στο παράδειγμά μας δημιουργήθηκε το στοιχείο ελέγχου με τίτλο «Πόλη». Σε αυτό αποδόθηκε αρχικά η προεπιλεγμένη τιμή την οποία μπορεί ο τελικός χρήστης να αλλάξει.

Στην προβολή αυτή μπορούμε να επεμβούμε, χρησιμοποιώντας μερικές από τις παρακάτω παραμέτρους του URL:

### ❖ user

Είναι η παράμετρος χρήστη. Εδώ πάντα χρησιμοποιούμε το email με το οποίο κάνουμε login στο σύστημα

- ❖ token  
Είναι ένα μοναδικό κλειδί που δίνει πρόσβαση στην υπηρεσία διαδραστικών χαρτών
- ❖ report  
Με το report επιλέγουμε τον χάρτη που επιθυμούμε να προβάλουμε
- ❖ lat  
Είναι το γεωγραφικό πλάτος. Αν το παραλείψουμε χρησιμοποιείται το 39
- ❖ lng  
Είναι το γεωγραφικό μήκος. Αν το παραλείψουμε χρησιμοποιείται το 23
- ❖ zoom  
Είναι η προεπιλεγμένη κλίμακα. Αν το παραλείψουμε χρησιμοποιείται το 7
- ❖ toolbar  
Με toolbar=0 δεν εμφανίζεται η εργαλειοθήκη στο πάνω μέρος
- ❖ zoomslider  
Με zoomslider=0 δεν εμφανίζεται η γραμμή κύλισης στο κάτω μέρος



Εικόνα 5.3.2 Παράδειγμα προβολής χωρίς εργαλειοθήκη και κλίμακα

Στο παραπάνω παράδειγμα, προβάλουμε τον χάρτη χωρίς εργαλειοθήκη και χωρίς γραμμή κύλισης κλίμακας. Επειδή στον χάρτη δεν συμπεριλάβαμε κάποια παράμετρο, εμφανίζεται να καλύπτει το σύνολο της οθόνης. Το URL είναι της μορφής :

```
http://servername:8079/m/?user=USEREMAIL&token=XXXXX&zoom=7&report=56&toolbar=0&zoomslider=0
```

## 5.4. Υπηρεσία Γεωγραφικών Δεδομένων REST API

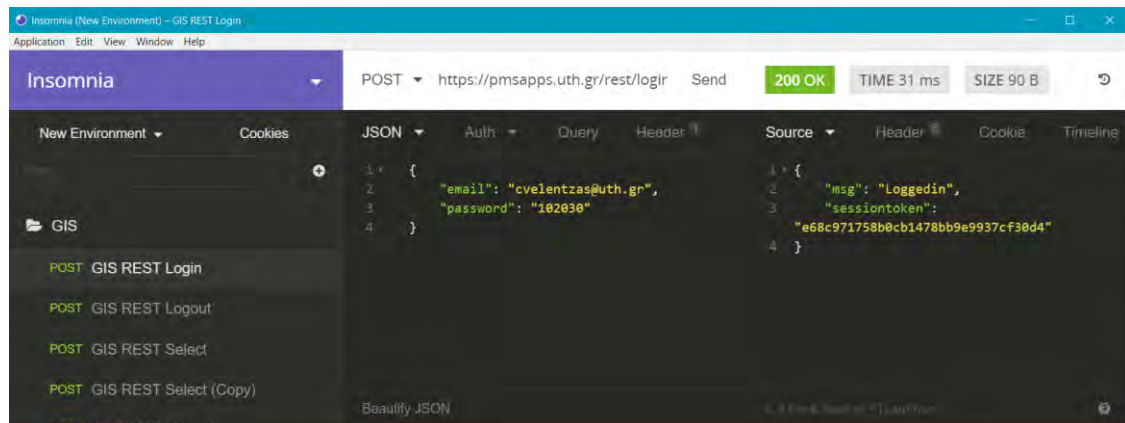
Για την θεωρείται η μεταπτυχιακή διατριβή πλήρης θα πρέπει να υποστηρίζει ένα πρωτόκολλο επικοινωνίας με εξωτερικές εφαρμογές, ευέλικτο και ανοιχτής αρχιτεκτονικής. Για αυτό το λόγο υλοποιήθηκε μια υπηρεσία τύπου REST.

Η υπηρεσία είναι ενεργοποιημένη στην διαδρομή <https://pmsapps.uth.gr/rest> [36]. Οι κλήσεις που μπορούμε να κάνουμε είναι μόνο τύπου Post με content-type=application/json, μιας και στηρίζεται στο JSON για την ανταλλαγή δεδομένων. Στο REST endpoint μας μπορούμε να κάνουμε login, data και logout ενέργειες:

❖ login

Κάνοντας μια post κλήση στην διεύθυνση <https://pmsapps.uth.gr/rest/login> μπορούμε να αποκτήσουμε ένα sessiontoken με το οποίο μας εξουσιοδοτεί η υπηρεσία για κλήσεις data. Με επιτυχημένο login μας επιστρέφει και ένα μήνυμα "msg":"Loggedin". Σε αντίθετη περίπτωση θα περιέχει το μήνυμα αποτυχίας πχ "msg":"credential error». Τα JSON δεδομένα που στείλαμε είναι:

```
Request https://pmsapps.uth.gr/rest/login
{
  "email": "cvelentzas@uth.gr",
  "password": "102030"
}
Και λάβαμε :
Response
{
  "msg": "Loggedin",
  "sessiontoken": "e68c971758b0cb1478bb9e9937cf30d4"
}
```



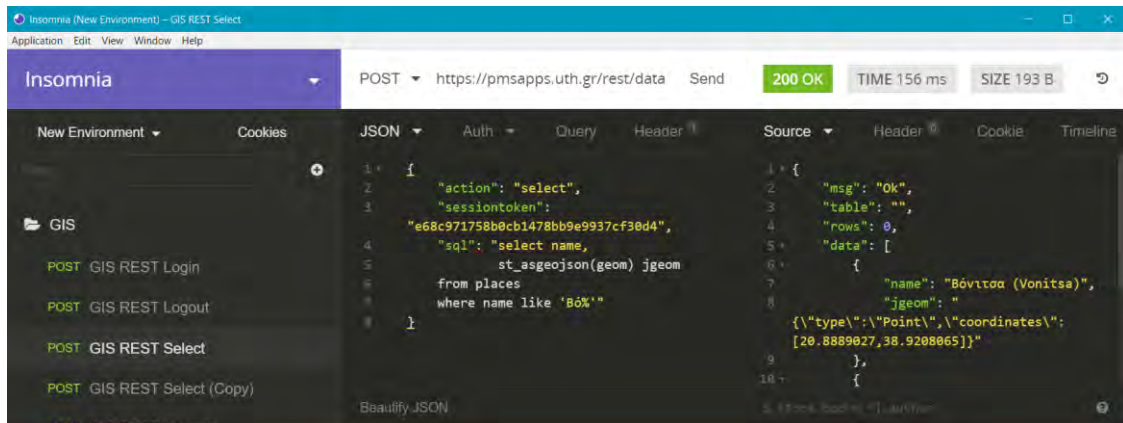
Εικόνα 5.4.1 Rest Login

❖ data

Αφού αποκτήσαμε το sessiontoken "e68c971758b0cb1478bb9e9937cf30d4" μπορούμε να εκτελέσουμε κλήσεις data. Για παράδειγμα θέλουμε να πάρουμε τα γεωγραφικά δεδομένα από τον πίνακα places για μέρη που αρχίζουν από «Βό».

```
Request https://pmsapps.uth.gr/rest/data
{
  "action": "select",
  "sessiontoken": "e68c971758b0cb1478bb9e9937cf30d4",
  "sql": "select name,
```

```
st_asgeojson(geom) jgeom
from places
where name like 'Βό%'
}
Response
{
  "msg": "Ok",
  "data": [
    {
      "name": "Βόνιτσα (Vonitsa)",
      "jgeom":
        "{\"type\":\"Point\",\"coordinates\":[20.8889027,38.9208065]}"
    },
    {
      "name": "Βόλος",
      "jgeom":
        "{\"type\":\"Point\",\"coordinates\":[22.9457579,39.3609876]}"
    }
  ]
}
```



Εικόνα 5.4.2 Rest Data

Στην κλήση data βλέπουμε ότι υπάρχουν οι παράμετροι action, sessiontoken και sql. Οι δυνατές τιμές που μπορεί να πάρει η παράμετρος action είναι οι εξής:

#### 1.select

Με την παράμετρο “select” κατευθύνουμε την υπηρεσία να εκτελέσει ένα ερώτημα. Μαζί με την select θα πρέπει να εισάγουμε και μια παράμετρο sql, όπως στο προηγούμενο παράδειγμα.

#### 2.Execute

Με την παράμετρο “execute” μπορούμε να εκτελέσουμε DML και DDL sql statements. Για παράδειγμα ας δημιουργήσουμε ένα πίνακα με όνομα “test” και πεδία

Id :Ακέραιος με αυτόματη αρίθμηση

Code :Κωδικός, κείμενο 50 χαρακτήρων





**Βελέντζας Πολυχρόνης**  
**Διαχείριση Γεωγραφικών Ερωτημάτων και**  
**Υπηρεσία Διαδραστικών Χαρτών για Ιστοσελίδες**

Descr :Περιγραφή, κείμενο 255 χαρακτήρων

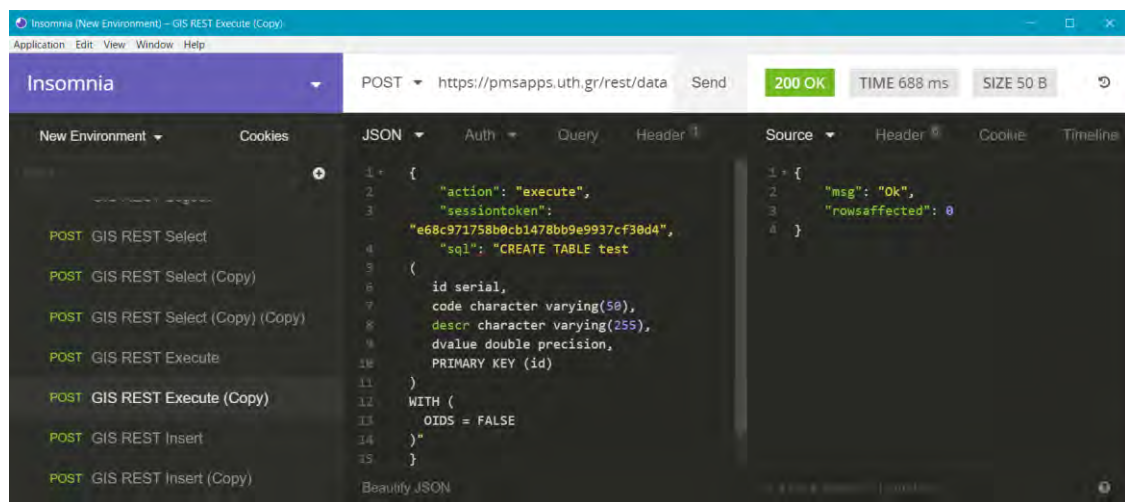
Rvalue:Πραγματικός αριθμός, διπλής ακρίβειας

Κάνουμε την παρακάτω κλήση :

```
Request https://pmsapps.uth.gr/rest/data
{
  "action": "execute",
  "sessiontoken": "e68c971758b0cb1478bb9e9937cf30d4",
  "sql": "CREATE TABLE test
(
  id serial,
  code character varying(50),
  descr character varying(255),
  dvalue double precision,
  PRIMARY KEY (id)
)
WITH (
  OIDS = FALSE
)"
}

Response
{
  "msg": "Ok",
  "rowsaffected": 0
}
```

Μετά από αυτή την κλήση, έχουμε δημιουργήσει τον πίνακά μας. Στην συνέχεια μπορούμε να του εισάγουμε δεδομένα ή να του σβήσουμε εγγραφές στέλνοντας το αντίστοιχο delete statement.



Εικόνα 5.4.3 Rest, δημιουργία πίνακα

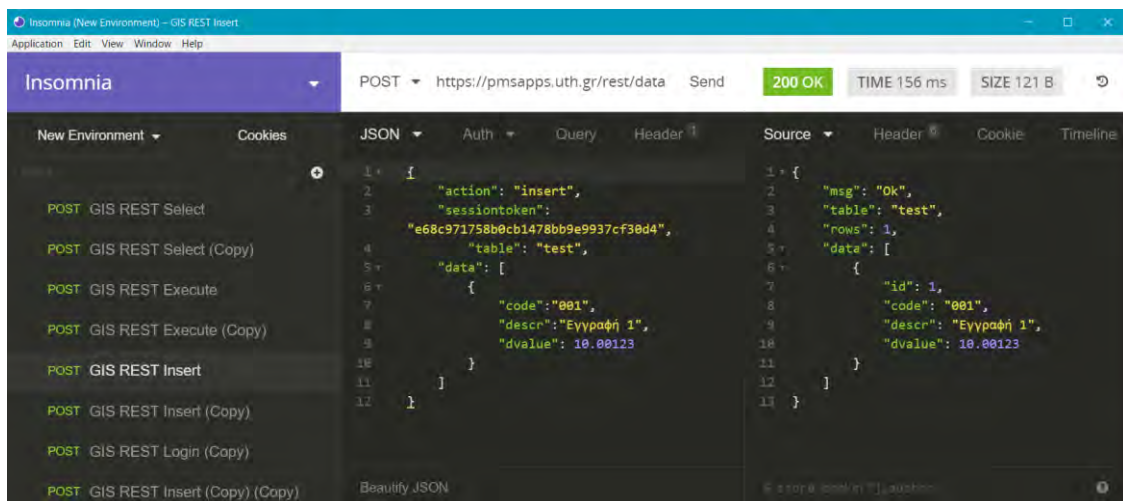
### 3.Insert

Με την παράμετρο “insert” μπορούμε με εύκολο τρόπο να εισάγουμε δεδομένα σε υπάρχοντα πίνακα της βάσης δεδομένων μας. Στο προηγούμενο παράδειγμα δημιουργήσαμε τον πίνακα test. Τώρα θα εισάγουμε μια νέα γραμμή στον πίνακά μας.

```
Request https://pmsapps.uth.gr/rest/data
{
  "action": "insert",
  "sessiontoken": "e68c971758b0cb1478bb9e9937cf30d4",
  "table": "test",
  "data": [
    {
      "code": "001",
      "descr": "Εγγραφή 1",
      "dvalue": 10.00123
    }
  ]
}

Response
{
  "msg": "Ok",
  "table": "test",
  "rows": 1,
  "data": [
    {
      "id": 1,
      "code": "001",
      "descr": "Εγγραφή 1",
      "dvalue": 10.00123
    }
  ]
}
```

Η υπηρεσία μας ενημερώνει ότι πραγματοποιήθηκε η εισαγωγή της εγγραφής και μας επιστρέφει όλα τα πεδία, ακόμη και το πεδίο “id” που έχει αυτόματη αρίθμηση.



Εικόνα 5.4.4 Εισαγωγή εγγραφής στον πίνακα

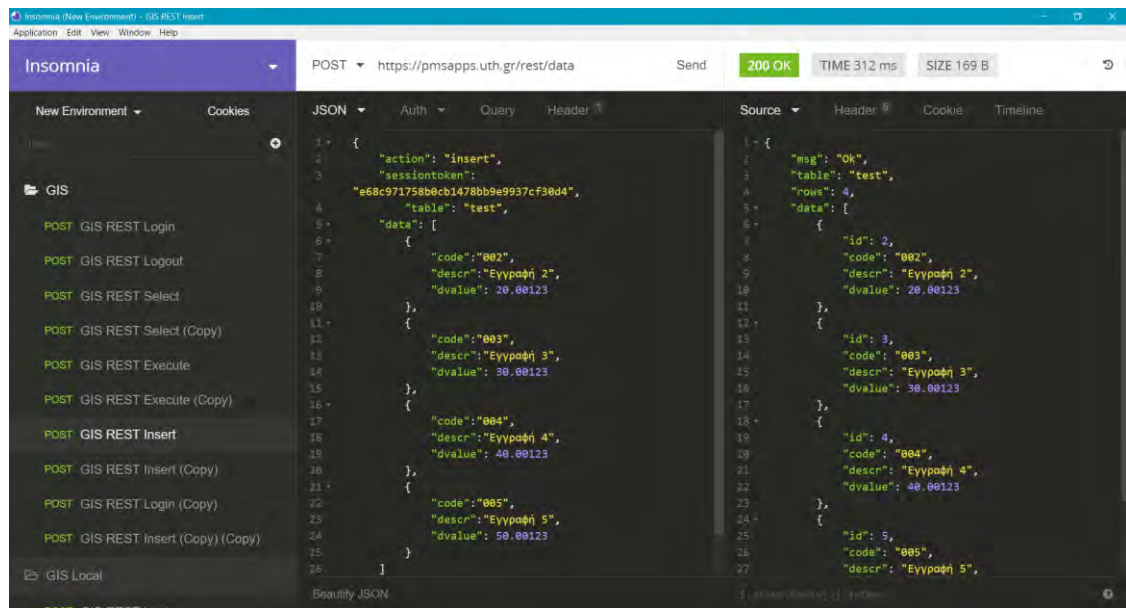
Επιπλέον θα μπορούσαμε να εισάγουμε περισσότερες από μια εγγραφές με μία κλήση. Στο επόμενο παράδειγμα εισάγουμε τέσσερις εγγραφές :

```
Request https://pmsapps.uth.gr/rest/data
{
  "action": "insert",
```



```
"sessiontoken": "e68c971758b0cb1478bb9e9937cf30d4",
"table": "test",
"data": [
  {
    "code": "002",
    "descr": "Εγγραφή 2",
    "dvalue": 20.00123
  },
  {
    "code": "003",
    "descr": "Εγγραφή 3",
    "dvalue": 30.00123
  },
  {
    "code": "004",
    "descr": "Εγγραφή 4",
    "dvalue": 40.00123
  },
  {
    "code": "005",
    "descr": "Εγγραφή 5",
    "dvalue": 50.00123
  }
]
}
```

```
Response
{
  "msg": "Ok",
  "table": "test",
  "rows": 4,
  "data": [
    {
      "id": 2,
      "code": "002",
      "descr": "Εγγραφή 2",
      "dvalue": 20.00123
    },
    {
      "id": 3,
      "code": "003",
      "descr": "Εγγραφή 3",
      "dvalue": 30.00123
    },
    {
      "id": 4,
      "code": "004",
      "descr": "Εγγραφή 4",
      "dvalue": 40.00123
    },
    {
      "id": 5,
      "code": "005",
      "descr": "Εγγραφή 5",
      "dvalue": 50.00123
    }
  ]
}
```

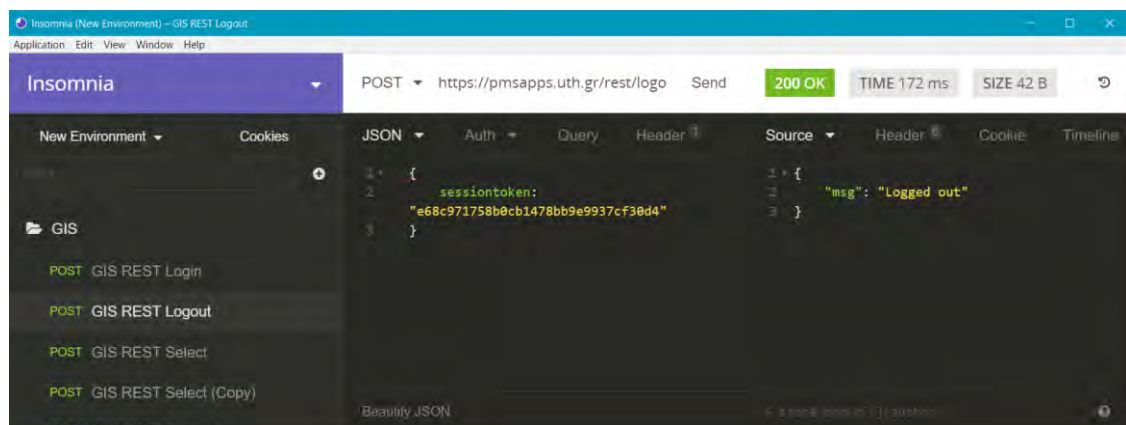


Εικόνα 5.4.3 Πολλές εισαγωγές με μια κλήση

#### ❖ Logout

Κάνοντας κλήση στην “Logout” αποσυνδεόμαστε από την υπηρεσία Rest και αποδεσμεύουμε το sessiontoken. Για παράδειγμα η παρακάτω κλήση μας αποσυνδέει:

```
Request https://pmsapps.uth.gr/rest/logo
{
  sessiontoken: "e68c971758b0cb1478bb9e9937cf30d4"
}
Response
{
  "msg": "Logged out"
}
```



Εικόνα 5.4.4 Αποσύνδεση από την υπηρεσία Rest




## 5.5. Εφαρμογή κινητού, Συλλογή δεδομένων GPS

Για τις ανάγκες της μεταπτυχιακής διατριβής δημιουργήθηκε μια εφαρμογή για κινητά τηλέφωνα και tablets, όπου συλλέγονται δεδομένα από τον αισθητήρα GPS της συσκευής και αποστέλλονται στην υπηρεσία REST API για αποθήκευση. Η εφαρμογή λειτουργεί σε όλα τα κινητά τηλέφωνα και tablet με λειτουργικά συστήματα Android και IOS (iPhone, iPad).

Η λειτουργία της είναι αρκετά απλή, αφού τρέξουμε την εφαρμογή “TrackLocation” ανοίγει το περιβάλλον της εφαρμογής και προτείνει τα Rest Endpoint (διεύθυνση υπηρεσίας REST), email, κωδικό και Αναγνωριστικό. Μπορεί κάποιος με το email του και διαφορετικό αναγνωριστικό κάθε φορά, να καταγράψει όσες συσκευές επιθυμεί και να τις παρακολουθήσει μέσω του συνδεδεμένου Rest Endpoint.

Αφού ξεκινήσει η εφαρμογή θα πρέπει να πατήσουμε το «Ενεργοποίηση» για να αρχίσει η συλλογή των στιγμάτων. Αν στο γεωγραφικό μήκος και πλάτος δεν εμφανίζεται ένδειξη, τότε θα πρέπει να ενεργοποιήσουμε το GPS της συσκευής γιατί είναι απενεργοποιημένο. Η εφαρμογή αποθηκεύει ένα στίγμα ανά λεπτό και ανά 10 λεπτά επικοινωνεί με το Rest endpoint για να αποστείλει τα στίγματα που εκκρεμούν. Στην περίπτωση που δεν υπάρχει διαθέσιμη σύνδεση στο διαδίκτυο, η εφαρμογή συγκρατεί στην μνήμη της όσα στίγματα δεν έχει στείλει. Όταν ξανασυνδεθεί αργότερα θα στείλει και όσα στίγματα εκκρεμούσαν. Δεδομένα η εφαρμογή μπορεί να στείλει είτε μέσω ασύρματης σύνδεσης WiFi είτε μέσω του παρόχου τηλεφωνίας με 3G, 4G και GPRS.

Κάθε φορά που αποστέλλει δεδομένα μας ενημερώνει με το πλήθος εγγγραφών που έστειλε. Στο παρακάτω παράδειγμα μας ενημέρωσε ότι έστειλε 9 εγγγραφές. Για να κλείσουμε την εφαρμογή μπορούμε να πατήσουμε το πλήκτρο «Έξοδος».



Ενεργοποίηση ☒

Rest Endpoint <https://pmsapps.uth.gr/rest>

E-mail [cvelentzas@uth.gr](mailto:cvelentzas@uth.gr)

Κωδικός ●●●●●●

Αναγνωριστικό [GPS Sensor 1](#)

Γεωγρ. Μήκος 39.364878

Γεωγρ. Πλάτος 22.931736

Αποστολή Ενημερώθηκαν 1 εγγραφές...

[Αποστολή Τώρα](#)

[Έξοδος](#)

Εικόνα 5.5.1 Εφαρμογή συλλογής στιγμάτων GPS





*Βελέντζας Πολυχρόνης*  
*Διαχείριση Γεωγραφικών Ερωτημάτων και*  
*Υπηρεσία Διαδραστικών Χαρτών για Ιστοσελίδες*

## 6.Αξιολόγηση και Συμπεράσματα

Στην παρούσα μεταπτυχιακή διατριβή προσπαθήσαμε να δημιουργήσουμε ένα σύστημα διαχείρισης γεωγραφικών ερωτημάτων και μια υπηρεσία διαδραστικών χαρτών για ιστοσελίδες. Αναπτύχθηκαν τέσσερις ξεχωριστές εφαρμογές που λειτουργούν συνδυαστικά. Θα αναλύσουμε ξεχωριστά την καθεμιά και στο τέλος θα αξιολογήσουμε συνολικά την πλατφόρμα.

Η βασική εφαρμογή της πλατφόρμας είναι η “Maps”. Στην εφαρμογή αυτή αναπτύχθηκε η «Ιστοσελίδα» και η «Σχεδίαση Γεωγραφικών Ερωτημάτων». Η ιστοσελίδα υποστηρίζει επαρκώς την λειτουργία της και δεν απαιτεί κάποια βελτίωση ή μελλοντική αλλαγή. Η σχεδίαση ερωτημάτων είναι αρκετά πλήρης, αλλά θα μπορούσε να βελτιωθεί στο σημείο των χαρτών, όπου προτείνεται μελλοντικά η υποστήριξη πολλών διαφορετικών πηγών χαρτών όπως οι παρακάτω, ενδεικτικά :

- ❖ AddisMap [37]
- ❖ Agilogik GmbH [38]
- ❖ Bexhill-OSM [39]
- ❖ BuioMetria Partecipativa [40]
- ❖ CartoDB [41]
- ❖ OpenTopoMap [42]
- ❖ Geofabrik Map [43]
- ❖ ITO Map [44]
- ❖ MapQuest [45]
- ❖ OpenSeaMap [46]
- ❖ OpenScienceMap [47]
- ❖ OpenStreetMap [6]

Ομοίως θα προτείνουμε να επεκταθεί και η εφαρμογή “MapsViewer” με τον ίδιο τρόπο. Η ενσωμάτωση των παραπάνω χαρτών θα έδινε πιο πλούσιο αποτέλεσμα στον τελικό χρήστη. Επιπλέον θα δινόταν η δυνατότητα επιλογής ενός δωρεάν χάρτη, με προφανή οφέλη. Μια ακόμη δυνατότητα που θα βελτιώνει το τελικό αποτέλεσμα είναι η ταυτόχρονη προβολή πολλών επιπέδων γεωγραφικών ερωτημάτων, με την δυνατότητα ο χρήστης να μπορεί να τα προσθαφαιρεί κατά βούληση.

Για την εφαρμογή “TrackLocation” το αποτέλεσμα ήταν ικανοποιητικό, μιας και ο λόγος υλοποίησης της ήταν υποστηρικτικός στην διατριβή και η λειτουργικότητά της σχετικά απλή. Δεν υπάρχει κάτι που θα μπορούσε να βελτιωθεί σε αυτή την περίπτωση.



Η υπηρεσία “RestService” πιστεύω καλύπτει τις ανάγκες επικοινωνίας εξωτερικών εφαρμογών και είναι υλοποιημένη με αρκετά ανοιχτή αρχιτεκτονική, δίνοντας την δυνατότητα εξαιρετικά ευέλικτης επικοινωνίας. Θα μπορούσε να βελτιωθεί μελλοντικά αν υιοθετούσε μερικά βοηθητικά χαρακτηριστικά όπως υποστήριξη σελιδοποίησης (pagination) και διαχείριση των πόρων με ολοκληρωμένη χρήση HTTP verbs.

Αφού αναφερθήκαμε αναλυτικά σε κάθε εφαρμογή θα πρέπει να εξετάσουμε και να αξιολογήσουμε την πλατφόρμα ως σύνολο. Ο σκοπός της πλατφόρμας είναι η ανάπτυξη κινητών και διαδικτυακών γεωγραφικών εφαρμογών. Καταφέραμε να δημιουργήσουμε τη γεωγραφική βάση δεδομένων όπου αποθηκεύουμε τα δεδομένα μας. Δημιουργήσαμε ένα περιβάλλον διαχείρισης της βάσης καθώς και των γεωγραφικών ερωτημάτων. Κατασκευάσαμε μια υπηρεσία που μας βοηθά στην προβολή των ερωτημάτων πάνω σε χάρτη, υποστηρίζοντας όλες τις σύγχρονες συσκευές με φυλλομετρητή και σύνδεση στο διαδίκτυο. Επιπλέον υποστηρίξαμε τις ανάγκες υποστήριξης εξωτερικών εφαρμογών τρίτων κατασκευαστών και κατασκευάσαμε μια εφαρμογή κινητών που συνεργάζεται με αυτή την υπηρεσία. Σαν σύνολο η πλατφόρμα έχει επιτύχει τον σκοπό της και για να παρουσιάσουμε τα προτερήματά της πρέπει να δούμε το συνολικό όφελος κατασκευής μιας εφαρμογής συλλογής και παρουσίασης γεωγραφικών δεδομένων, με και χωρίς την βοήθεια της πλατφόρμας.

Παρακάτω παρουσιάζουμε ένα συγκριτικό πίνακα, για την ανάπτυξη επιμέρους εφαρμογών και προγραμμάτων από τα οποία ο προγραμματιστής επωφελείται:

1. Χωρίς την βοήθεια της πλατφόρμας
2. Ελάχιστη αξιοποίηση της πλατφόρμας, χρησιμοποιώντας την βάση δεδομένων και την υπηρεσία Rest
3. Εξολοκλήρου, χρησιμοποιώντας την πλατφόρμα

Υλοποίηση	Βάση Δεδομένων	Web Server	Rest	Ιστοσελίδα	Προβολή σε συσκευές
1.	✗	✗	✗	✗	✗
2.	✓	✓	✓	✗	✗
3.	✓	✓	✓	✓	✓

Παρατηρώντας τον παραπάνω πίνακα είναι προφανές ότι στην περίπτωση μη χρησιμοποίησης της πλατφόρμας θα πρέπει να αναπτυχθούν από μηδενική βάση τα:

- ❖ Βάση Δεδομένων
- ❖ Web Server
- ❖ Rest
- ❖ Ιστοσελίδα
- ❖ Προβολή σε συσκευές



Πρόκειται για μια χρονοβόρα και κοστοβόρα διαδικασία που κάποιος δεν θα επιλέξει να ακολουθήσει, πλην της περίπτωσης μιας πολύ ειδικής περίπτωσης που δεν θα μπορούσε να καλυφθεί από τις δυνατότητες της πλατφόρμας μας.

Ακόμη και στην δεύτερη περίπτωση, που κάποιος δεν θα ήθελε να χρησιμοποιήσει την ιστοσελίδα και την προβολή σε συσκευές που προσφέρει η πλατφόρμα, το όφελος δεν είναι αμελητέο. Η περίπτωση που θα επέλεγε ένας προγραμματιστής αυτή την λύση θα ήταν για να αποκρύψει την λειτουργία της πλατφόρμας από τον τελικό χρήστη, παρουσιάζοντας έτσι την λύση του περισσότερο ανεξάρτητη και αυτόνομη.

Στην Τρίτη περίπτωση η ανάπτυξη της εφαρμογής θα είχε πολύ μικρό χρόνο υλοποίησης και θα ήταν πολύ αποτελεσματική ως προς το τελικό αποτέλεσμα.

Συνολικά η πλατφόρμα παρουσιάζεται αρκετά πλήρης σε χαρακτηριστικά αλλά μέχρι στιγμής απευθύνεται μόνο στο Ελληνικό κοινό. Όλη η διαχείρισή της και οι εφαρμογές εμφανίζονται στην Ελληνική γλώσσα και δεν θα μπορεί να αξιοποιηθεί από κάποιον που δεν γνωρίζει Ελληνικά. Είναι επιβεβλημένη λοιπόν η ανάγκη, μελλοντικά να υποστηριχθούν και άλλες γλώσσες οδηγώντας στην «διεθνοποίηση» της πλατφόρμας.



*Βελέντζας Πολυχρόνης*  
*Διαχείριση Γεωγραφικών Ερωτημάτων και*  
*Υπηρεσία Διαδραστικών Χαρτών για Ιστοσελίδες*



## Αναφορές και Βιβλιογραφία

---

- [1] "MapServer," [Online]. Available: <http://mapserver.org>. [Accessed 10 6 2017].
- [2] "uDig," [Online]. Available: <http://udig.refractory.net>. [Accessed 10 6 2017].
- [3] "GRASS," [Online]. Available: <https://grass.osgeo.org/documentation/general-overview/>. [Accessed 10 6 2017].
- [4] "Esri-arcgis," [Online]. Available: <http://gisgeography.com/esri-arcgis-software-review-guide/>. [Accessed 10 6 2017].
- [5] "QGIS," [Online]. Available: <http://www.qgis.org/en/site/>. [Accessed 10 6 2017].
- [6] "OpenStreetMap," [Online]. Available: <http://www.openstreetmap.org/>. [Accessed 15 6 2017].
- [7] "Slippy Map," OpenStreetMap, [Online]. Available: [http://wiki.openstreetmap.org/wiki/Slippy\\_Map](http://wiki.openstreetmap.org/wiki/Slippy_Map). [Accessed 15 6 2017].
- [8] "Leaflet," OpenStreetMap, [Online]. Available: <http://wiki.openstreetmap.org/wiki/Leaflet>. [Accessed 15 6 2017].
- [9] "iD," OpenStreetMap, [Online]. Available: <http://wiki.openstreetmap.org/wiki/iD>. [Accessed 15 6 2017].
- [10] "SPA," [Online]. Available: [https://en.wikipedia.org/wiki/Single-page\\_application](https://en.wikipedia.org/wiki/Single-page_application). [Accessed 10 6 2017].
- [11] M. A. S. R. Leonard Richardson, RESTful Web APIs, O'Reilly Media , 2013.
- [12] "JSON," [Online]. Available: <http://www.json.org/>. [Accessed 10 6 2017].
- [13] "PostgreSQL," [Online]. Available: <https://en.wikipedia.org/wiki/PostgreSQL>. [Accessed 10 6 2017].
- [14] A. F. A. S. Ibrar Ahmed, PostgreSQL Developer's Guide, Packt Publishing, 2015.
- [15] "PostGIS," [Online]. Available: <https://en.wikipedia.org/wiki/PostGIS>. [Accessed 20 6 2017].





- [16] A. Marquez, PostGIS Essentials, Packt Publishing, 2015.
- [17] "Embarcadero," Embarcadero, [Online]. Available: <https://www.embarcadero.com>. [Accessed 10 6 2017].
- [18] M. Cantu, Object Pascal Handbook (updated for Delphi 10.1 Berlin) [1 ed.], Marco Cantù, 2016.
- [19] "Delphi," [Online]. Available: [https://en.wikipedia.org/wiki/Delphi\\_\(programming\\_language\)](https://en.wikipedia.org/wiki/Delphi_(programming_language)). [Accessed 6 10 2017].
- [20] "Delphi Cookbook - Second Edition," [Online]. Available: <https://www.packtpub.com/application-development/delphi-cookbook-second-edition>. [Accessed 10 6 2017].
- [21] "Embarcadero - FireDAC," [Online]. Available: <https://www.embarcadero.com/products/rad-studio/firedac>. [Accessed 10 6 2017].
- [22] "Indy," [Online]. Available: <http://www.indyproject.org/index.en.aspx>.
- [23] "uniGUI," FMSoft, [Online]. Available: <http://unigui.com/>. [Accessed 10 6 2017].
- [24] "ISAPI," [Online]. Available: [https://en.wikipedia.org/wiki/Internet\\_Server\\_Application\\_Programming\\_Interface](https://en.wikipedia.org/wiki/Internet_Server_Application_Programming_Interface).
- [25] "Sencha Ext Js," [Online]. Available: <https://www.sencha.com/products/extjs/#overview>. [Accessed 10 6 2017].
- [26] "Sencha Touch," Sencha, [Online]. Available: <https://www.sencha.com/products/touch/#overview>. [Accessed 10 6 2017].
- [27] R. Soni, Nginx: From Beginner to Pro, Apress.
- [28] "NGINX," [Online]. Available: <http://nginx.org/>. [Accessed 10 6 2017].
- [29] "Google Maps," Google, [Online]. Available: <https://www.google.com/maps>. [Accessed 10 6 2017].
- [30] "Google Maps API," Google, [Online]. Available: <https://developers.google.com/maps/>. [Accessed 10 6 2017].
- [31] "Android Studio," Google, [Online]. Available: <https://developer.android.com/studio/index.html>. [Accessed 10 6 2017].



- [32] "Google Developers," Google, [Online]. Available:  
<https://console.developers.google.com>. [Accessed 10 6 2017].
- [33] "Insomnia," Insomnia, [Online]. Available: <https://insomnia.rest/>. [Accessed 10 6 2017].
- [34] "Διαχείριση Γεωγραφικών Ερωτημάτων και Υπηρεσία Διαδραστικών Χαρτών για Ιστοσελίδες," Πανεπιστήμιο Θεσσαλίας, [Online]. Available: <https://pmsapps.uth.gr>. [Accessed 10 6 2017].
- [35] "Esri," [Online]. Available:  
<https://www.esri.com/library/whitepapers/pdfs/shapefile.pdf>. [Accessed 10 6 2017].
- [36] "Υπηρεσία Rest," Πανεπιστήμιο Θεσσαλίας, [Online]. Available:  
<https://pmsapps.uth.gr/rest>. [Accessed 10 6 2017].
- [37] "AddisMap," [Online]. Available: <http://www.addismap.com>. [Accessed 10 6 2017].
- [38] "Agilogik GmbH," [Online]. Available:  
[http://www.agilogik.de/portfolio\\_mapserver.shtml](http://www.agilogik.de/portfolio_mapserver.shtml). [Accessed 10 6 2017].
- [39] "Bexhill-OSM," [Online]. Available: <http://bexhill-osm.org.uk/>. [Accessed 10 6 2017].
- [40] "BuioMetria Partecipativa," [Online]. Available: <http://www.lucadelu.org/bmp/>. [Accessed 10 6 2017].
- [41] "CartoDB," [Online]. Available: <https://carto.com/location-data-services/basemaps/>.
- [42] "OpenTopoMap," [Online]. Available: <https://opentopomap.org>.
- [43] "Geofabrik," [Online]. Available: <http://tools.geofabrik.de/map>.
- [44] "ITO Map," [Online]. Available: [http://wiki.openstreetmap.org/wiki/Ito\\_Map](http://wiki.openstreetmap.org/wiki/Ito_Map).
- [45] "MapQuest," [Online]. Available: <http://wiki.openstreetmap.org/wiki/MapQuest>.
- [46] "OpenSeaMap," [Online]. Available:  
[http://openseamap.org/index.php?id=openseamap&no\\_cache=1](http://openseamap.org/index.php?id=openseamap&no_cache=1).
- [47] "OpenScienceMap," [Online]. Available: <http://www.opensciencemap.org/>.